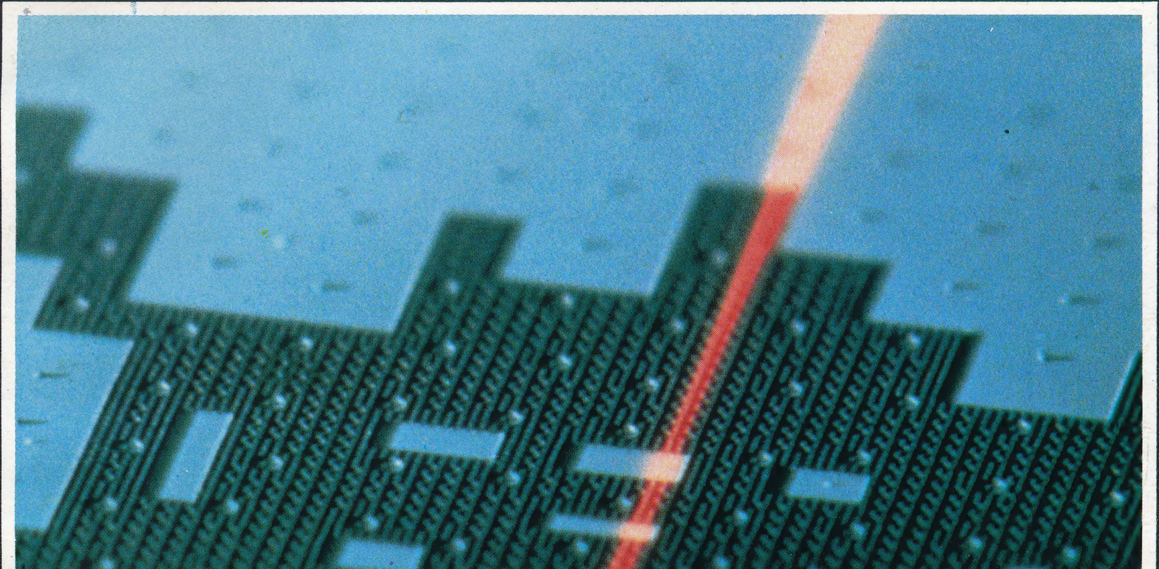
I.T.E.C.

Information
Technology
Electronics
Computers



46
in weekly parts

THE HOME STUDY COURSE IN ELECTRONICS AND COMPUTERS

U.K. 99p Australia $2.50 New Zealand $3.00 South Africa R2.75

Contents Part 46

COMMUNICATIONS – 16
Protocols – 1 1441

COMPUTERS & SOCIETY – 14
Personal computers 1449

BASIC THEORY REFRESHER
Transmission lines 1461

MICROPROCESSORS – 11
8-bit microprocessors 1464

I.T.E.C. is complete in 51 weekly parts (50 parts + Index).

POSTAL SUBSCRIPTIONS

UK readers

You can arrange to receive weekly copies of *I.T.E.C.* at 99p each inclusive of postage and packing. Simply calculate the cost of your subscriptions by multiplying the number of parts you require by 99p and send a cheque or postal order for that amount to:–

Subscriptions Dept., GEJ Publishing Ltd
187 Oxford Street, LONDON W1R 1AJ.

with a note of your name and address, where you wish the copies to be sent and the part number from which you want the subscription to start. Please allow 28 days for your subscription to be processed plus postal time.

Eire readers

As above, but you should calculate the cost of your subscription at IR£1.75 per part.

Other overseas readers

May pay in the currency of their choice provided that its value when converted to sterling is £1.50 per part. Minimum subscription 10 parts.

CONSULTANTS

This series has been produced in collaboration with **TEXAS INSTRUMENTS** Learning Centre. Editorial and graphics assistance is gratefully acknowledged.

U.K. Technical Consultant

Dr. Robert King, Reader in Communications Engineering, Imperial College of Science and Technology, London.

BINDERS

U.K. & Eire

Details of how to obtain your binders for *I.T.E.C.* will appear in part 50.

Australia

Write to *I.T.E.C.* Binders, Gordon & Gotch Ltd., PO Box 213, Alexandria, New South Wales 2015.

New Zealand

Write to *I.T.E.C.* Binders, Gordon & Gotch (NZ) Ltd., PO Box 1595, Wellington

South Africa

Binders are available at R7.95 (please add sales tax) from any branch of Central News Agency. In case of difficulty please write to Intermag, P.O. Box 57934, Springfield 2137.

Other countries

Binders are available from your local newsagent.

BACK NUMBERS

U.K.

Back numbers can be ordered from your newsagent or from *I.T.E.C.* BACK NOS., Whinfrey Strachan Ltd., 187 Oxford Street, London W1R 1AJ. Price 99p each inc. p + p. Please allow 21 days for delivery.

Eire

As for U.K. but please remit in Sterling if ordered by post.

Other countries

Copies available from your local newsagent.

INDEX

The final part of *I.T.E.C.* will be a fully cross-referenced INDEX to the complete work

Editor:
Amanda Kirk

Technical Editor:
Keith Brindley

Art Editor:
Alistair Carlisle

Sub Editor:
Nicholas Bellenberg

Copy Consultant:
Chris Wallace (Creative Technical Services)

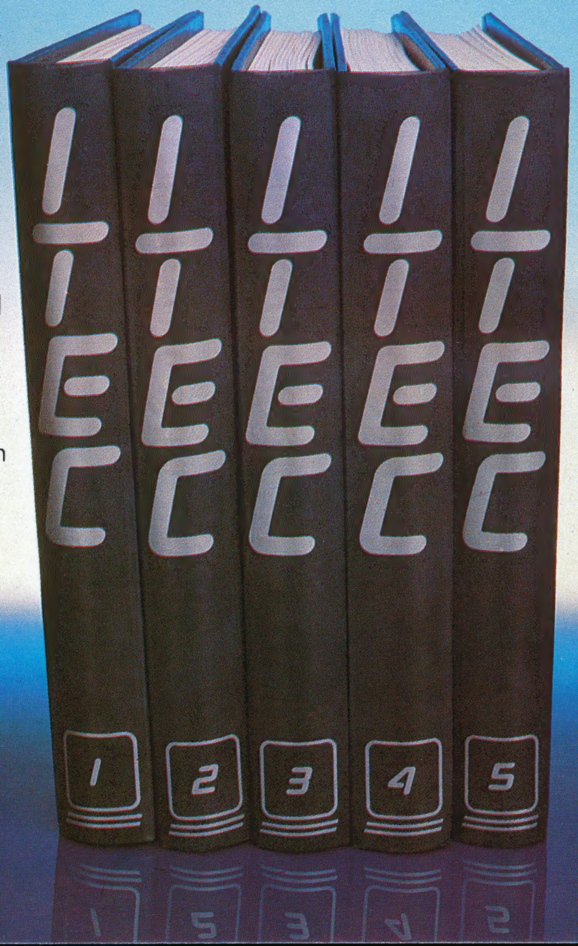
Published by GEJ Publishing Ltd.

Distributed by Whinfrey Strachan Ltd.
187 Oxford Street, London
W1R 1AJ

Printed in England by Southernprint Ltd.

Photographs, unless otherwise stated, are from the library of Texas Instruments.

Cover: high speed laser scanning system. (Photo: IBM).



16
COMMUNICATIONS

Protocols-1

The purpose of recommendations

In the same way that recommendations are provided by the CCITT for data transmission over analogue circuits (the V-series), recommendations are also provided for data transmission over digital networks. These recommendations form what is known as the **X-series**. Some have a direct mapping with those of the V-series: for example the DTE/DCE interface should follow recommendation V.24 on analogue networks, and recommendation **X.24** on digital networks.

Like its analogue equivalent, X.24 specifies a number of interchange circuits to be used by DTE and DCE. These are fewer in number and simpler than V.24 interchange circuits because in a purely digital interface, control functions are often performed by particular data codes rather than, say, separate interchange circuits. Interchange circuits recommended in X.24 are listed in *table 1*.

When DTE are connected to analogue networks, the interchange circuits recommended in V.24 may be used in a number of different ways to interface DTE with modems – the DCEs. We saw examples in *Communications 12* and *13* of 300 bits s^{-1} modems (following recommendation V.21), 1200/75 bits s^{-1} modems (V.23), 2400 bits s^{-1} modems (V.21 bis) and 48,000 bits s^{-1} modems (V.35 and V.36).

The X-series includes similar recommendations concerned with DTE/DCE interface onto digital networks. **X.20**, for example, recommends a general purpose DTE/DCE interface to allow the connection of asynchronous equipment to public digital data networks. The interchange circuits relating to X.20 interfaces are listed in *table 2*.

Table 3 lists the interchange circuits concerning recommendation **X.21** – a general purpose synchronous DTE/DCE interface to public digital data networks.

Recommendations X.20 and X.21 form the two basic interfaces used to connect DTE to digital data networks. The CCITT, recognising the fact that existing DTE only have a V.24 interface, also recommend **X.20 bis** (i.e. the second version of X.20) and **X.21 bis** (the second version of X.21). Interfaces following recommendations X.20 bis and X.21 bis use V.24-type interchange circuits between DTE and DCE, so the DCE appears as a modem to the DTE. X.20 bis and X.21 bis interfaces allow asynchronous and synchronous data transmission between DTE

Table 1
Recommendation X.24 interchange circuits

Circuit	Name	Direction
G	Signal ground or common return	—
Ga	DTE common return	to DCE
Gb	DCE common return	to DTE
T	Transmit	to DCE
R	Receive	to DTE
C	Control	to DCE
I	Indication	to DTE
S	Signal element timing	to DTE
B	Byte timing	to DTE
	cable screen	

Table 2
Recommendation X.20 for the DTE/DCE interface

Circuit	Name	Direction
G	Signal ground or common return	—
Ga	DTE common return	to DCE
Gb	DCE common return	to DTE
T	Transmit	to DCE
R	Receive	to DTE

following recommendation V.24 and digital data networks, but do not offer all the facilities of X.20 and X.21 interfaces.

Packet switched networks

The recommendation which specifies the DTE/DCE interface onto packet switched networks (see *Communications 15*) is recommendation **X.25**. For this reason, packet switched data networks such as Packet SwitchStream are often called **X.25 networks**, although recommendation X.25 refers specifically to the interface between packet terminals and the network.

Table 3
Recommendation X.21 interchange circuits

Circuit	Name	Direction
G	Signal ground or common return	—
Ga	DTE common return	to DCE
T	Transmit	to DCE
R	Receive	to DTE
C	Control	to DCE
I	Indication	to DTE
S	Signal element timing	to DTE
B	Byte timing	to DTE
	Cable screen	

Table 4
Comparison of X and V-series CCITT recommendations

X-series	
X.1	International user classes of service in public data networks
X.2	International user facilities in public data networks
X.3	Packet assembly/disassembly facility in a public data network
X.20	Interface between data terminal equipment and data circuit terminating equipment for start-stop transmission services on public data networks
X.20 bis	V.21 compatible interface between data terminal equipment and data circuit terminating equipment for start-stop transmission services on public data networks
X.21	General purpose interface between data terminal equipment and data circuit terminating equipment for synchronous operation on public data networks
X.21 bis	Use on public data networks of data terminal equipments which are designed for interfacing synchronous V-series modems
X.24	List of definitions of interchange circuits between data terminal equipment and data circuit terminating equipment on public data networks
X.25	Interface between data terminal equipment and data circuit terminating equipment for terminals operating in the packet mode on public data networks
X.28	DTE/DCE interface for a start-stop-mode data terminal equipment assessing the packet assembly/dissembly facility (PAD) on a public data network situated in the same country
X.29	Procedures for exchange on control information and user data between a packet mode DTE and a packet assembly/dissembly facility (PAD)
X.75	Terminal and transit call control procedures and data transfer systems on international circuits between packet switched data networks
X.121	International numbering plan for public data networks

V-series	
V.3	International Alphabet No 5 (ASCII)
V.4	General structure of signals of international Alphabet No 5 code
V.5	Standardisation of data signalling rates for synchronous data transmission in the general switched telephone network
V.6	Standardisation of data signalling rates for synchronous data transmission on leased telephone-type circuits
V.15	Use of acoustic coupling for data transmission
V.16	Medical analogue data transmission modems
V.19	Modems for parallel data transmission using telephone signalling frequencies
V.20	Parallel data transmission modems standardised for universal use in the general switched telephone network
V.21	300 bits s^{-1} modem standardised for use in the general switched telephone network
V.23	1200/75 bits s^{-1} modem standardised for use in the general switched telephone network
V.24	List of definitions for interchange circuits between data-terminal equipment and data circuit terminating equipment
V.25	Automatic calling and/or answering on the general switched telephone network including disabling or echo-suppressors on manually established calls
V.35	Data transmission at 48,000 bits s^{-1} using 60 to 108 kHz group band circuits
V.36	Modems for synchronous data transmission using 60-108 kHz group band circuits
V.56	Comparative tests of modems for use over telephone-type circuits
V.57	Comprehensive data test set for high data signalling rates

Asynchronous terminals, and for that matter synchronous terminals which are not packet terminals, must be connected to a packet switched network using packet assembler/dissembler DCE (PADs). This interface is specified by two recommendations: **X.28**, which covers the required interchange circuits; and **X.29**, which covers the control and data signalling procedures in the DTE/DCE interface. The complete packet assembly/dissembly facility is specified in recommendation **X.3**.

The functions of recommendations

Some of the more important recommendations regarding data communications in both the X-series and the V-series, are shown in *table 4*. New recommendations evolve as new techniques for data transmission are developed. Their main aim is to ensure that equipment connected to a data network can at least communicate with other hardware on the network. For example, if a personal computer signals to a modem using V.24 interchange circuit 105 (request to send), the modem must know that it wishes to transmit data. The actual data transmitted is dependent on other factors, such as software.

Similarly, on an X.25 data network, this hardware or electrical level of communication is satisfactorily met by equipment following recommendation X.21. Again, the form of the transmitted data is not specified and depends on software.

We can see therefore, that a set of rules is necessary between devices at some higher level than the primary hardware connection to enable useful communications. These sets of rules are known as **protocols**. Protocols are equipment or system dependent and this presents problems when interconnecting systems. At the **interface** between two systems, the protocols governing that interface must be followed in order for data to be exchanged.

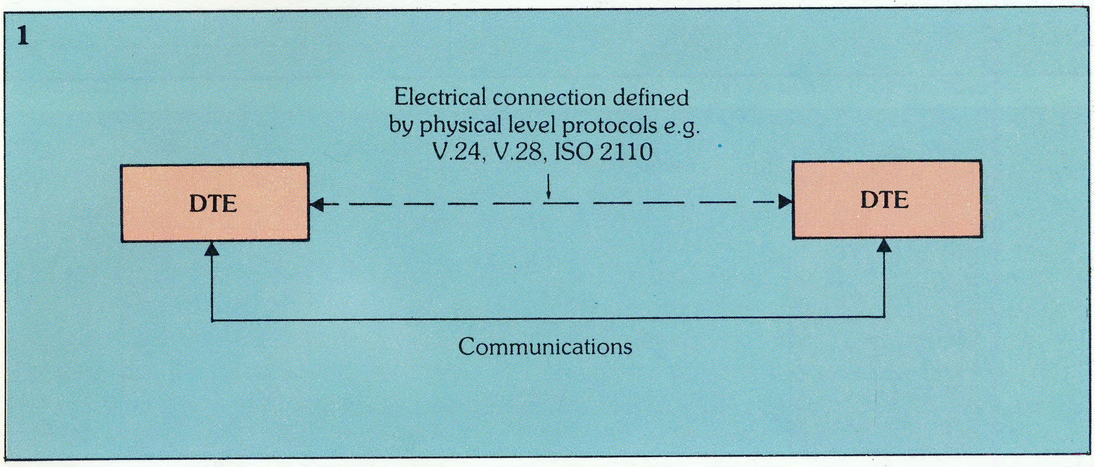
We have already seen simple examples of protocols and interfaces: the DTE/DCE interface between a terminal and the PSTN – the modem; and CCITT recommendation V.24 (together with V.28, ISO 2110 and V.3 – ASCII code) is the simplest and most straightforward example.

However, the complexity of the communications systems involved defines the complexity of the protocols used: for example, the terminal modem interface between digital equipment and the PSTN.

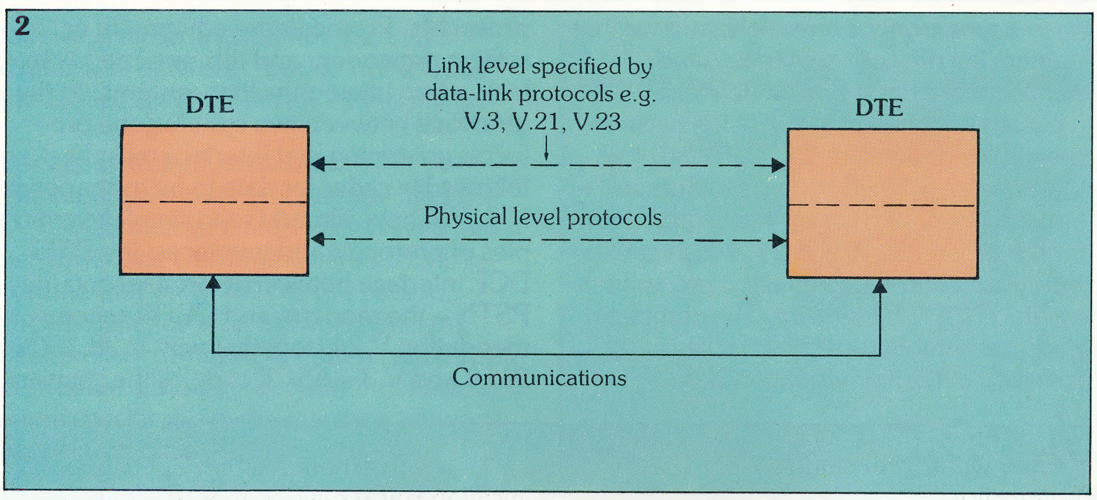
Recommendation V.24 defines the lowest **level** of the communications protocol. If a terminal and a modem both follow the V.24 (and V.28 and ISO 2110) recommendation, then equipment users know that the two devices may be connected together with a standard cable (which also follows the recommendations). Terminal and modem will then be able to communicate control, timing and data signals to each other, using the interchange circuits of recommendation V.24. This lowest level of communication is also known as the **electrical level**, or **physical level**, and is shown for our example in *figure 1*.

Even though terminal and modem may now communicate at the electrical level, the data which is exchanged between terminals, via the modem and PSTN, will not be of any use unless both

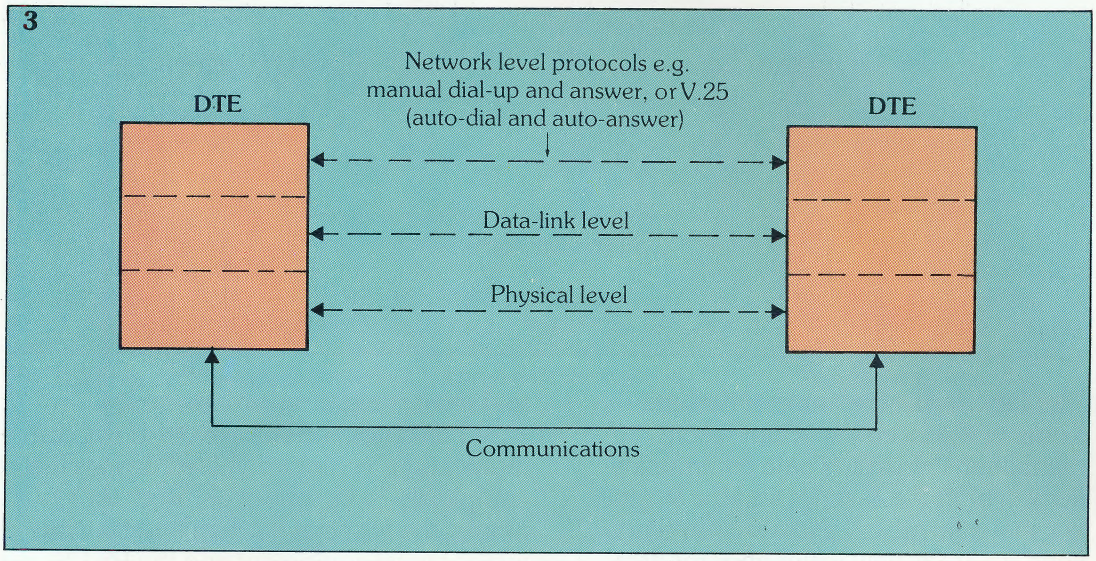
1


Electrical connection defined
by physical level protocols e.g.
V.24, V.28, ISO 2110
DTE
DTE
Communications

1. The lowest level of communication is known as the electrical level, or physical level.


2
DTE
Link level specified by data-link protocols e.g. V.3, V.21, V.23
DTE
Physical level protocols
Communications

2. The data-link and physical levels are the only two levels of protocol required for communication between two DTEs over a private leased line.


3
DTE
Network level protocols e.g. manual dial-up and answer, or V.25 (auto-dial and auto-answer)
DTE
Data-link level
Physical level
Communications

3. If data transmission is required over the PSTN, then a third or network level protocol is necessary.

terminals use the same **data protocol**. So, we see that a second, *higher* level protocol is required at the **data level**, or **link level** (also known as the **data-link level**).

The use of a standard data code (ASCII) specified in recommendation V.3, fulfills this requirement between the terminals and their respective modems. The two modems are connected by analogue PSTN lines, so coding the data in ASCII form is not suitable, of course. One of the modem transmission methods using audio frequency signals recommended in the CCITT V-series must therefore be used. The examples we have seen of recommendations V.21, V.22 bis, or V.23 are suitable for this purpose. *Figure 2* illustrates how these two levels are used to communicate between terminals on the PSTN.

Data-link and physical level protocols are the only two required if data is to be transmitted over a leased, private line between two DTEs. In such an application, the modems are permanently connected and one DTE communicates only with the other. If, on the other hand, a dialled-up connection over the PSTN is to be used, another protocol level is needed to route the connection over the PSTN as the call is made. In our example, this third level may be undertaken by one person dialling a telephone terminal number which routes the call through to the required terminal.

This third or **network level** may be implemented automatically, with auto-dial and auto-answer modems specified by recommendation V.25. *Figure 3* shows how the network level protocol is fulfilled by both the manual dial-up and answer method, and the auto-dial and auto-answer network solution.

A possible network with connected

DTE and DCE is shown in *figure 4*: the three levels are marked. Boundaries between levels are generally not so easily distinguished in a practical communications system, nevertheless this gives an adequate idea of their inter-relationships.

There are two points worthy of note in the network shown in *figure 4*:

1) The protocol levels of both DTE/DCE combinations need not be the same. For instance, DTE 1 may be connected to DCE 1 with a standard electrical, physical level connection, conforming to recommendation V.24, but DTE 2 may be connected to DCE 4 with a non-standard physical level connection.

Similarly, DCE 1 may be connected to DCE 2 with a V.21-type modem interface (a data level protocol), but DCE 3 may be connected to DCE 4 with a V.23-type interface.

2) Each level of protocol is dependent on the protocols at the lower levels, because any protocol uses the features and facilities provided by the lower level protocols.

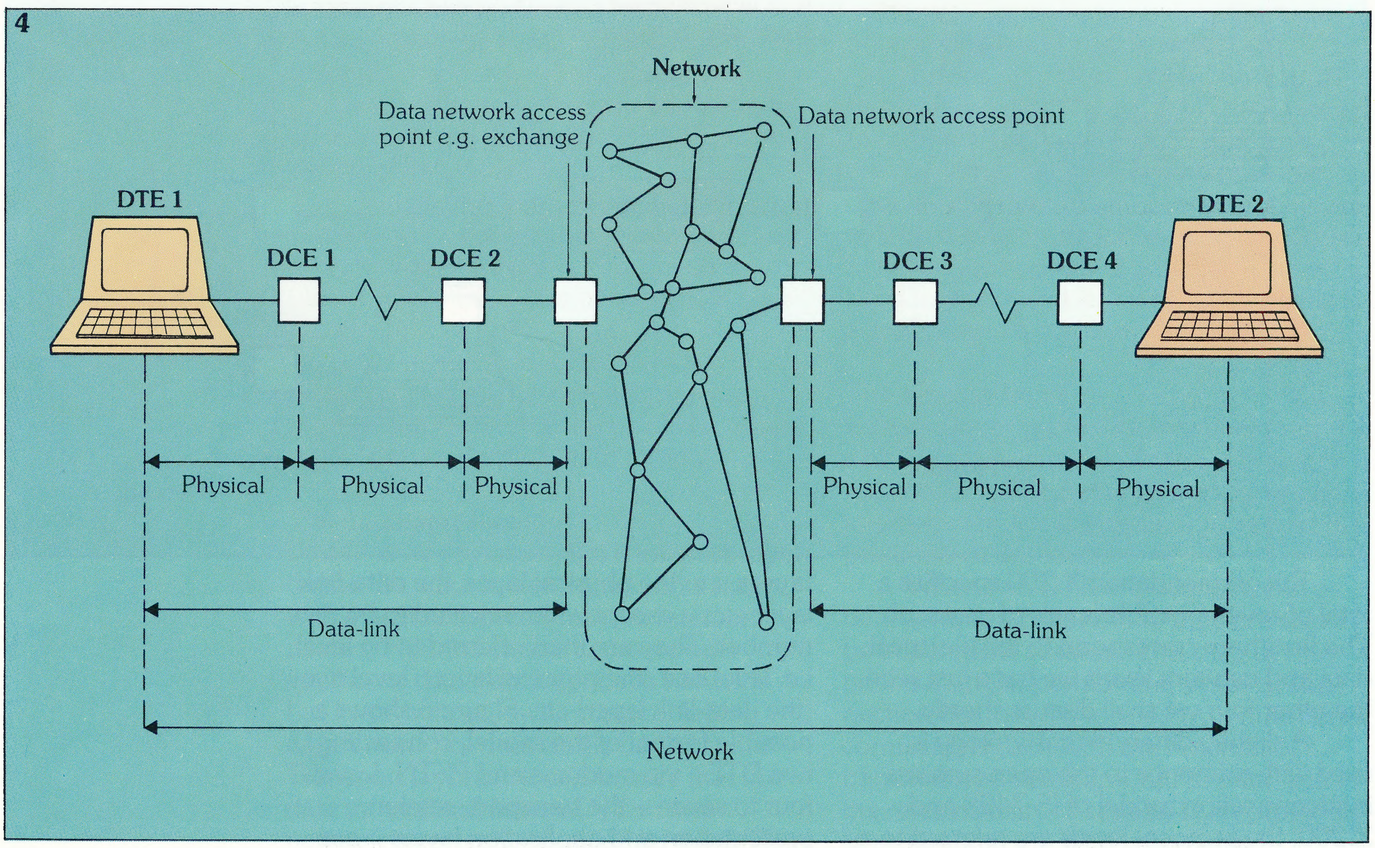
Three levels of protocol are sufficient for this application, but as we shall see, more levels are often required.

Open systems

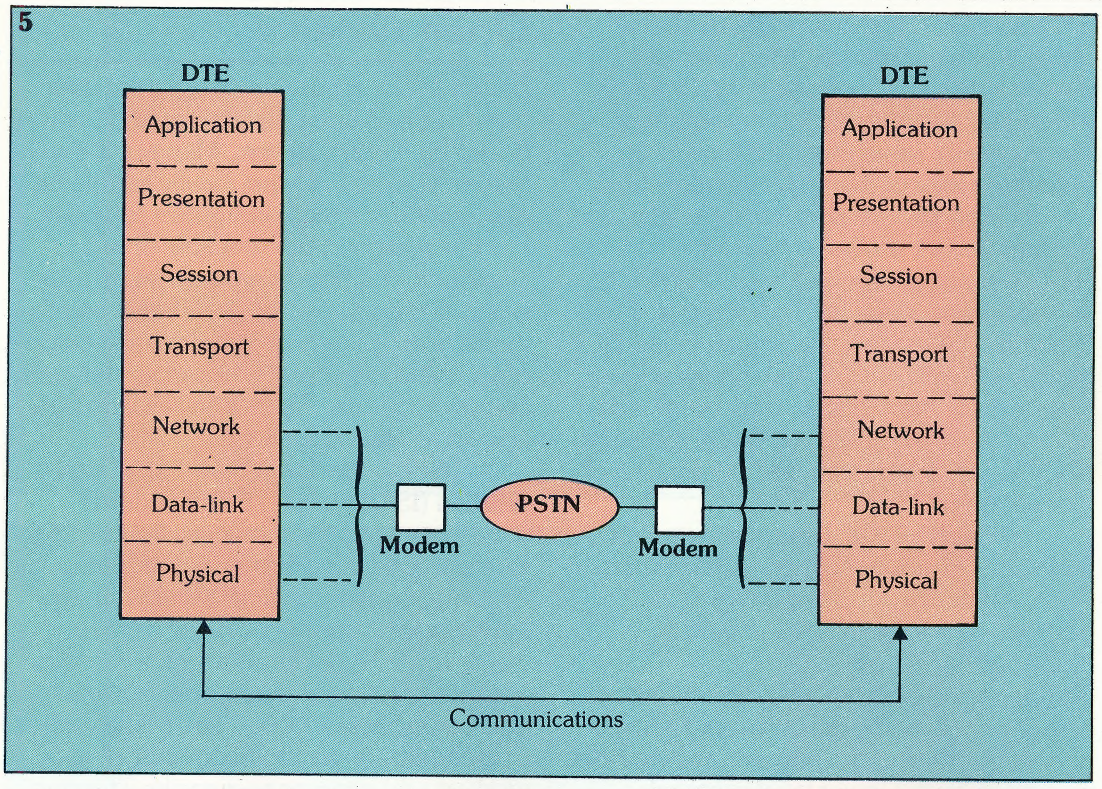
Before 1977, a number of different data networks had been developed and produced by manufacturers. Many of these networks were sold specifically to link the manufacturer's own computers together. For this reason, there was little or no likelihood of different manufacturer's networks being connected together. The protocols used, therefore, were usually autonomous and often prevented inter-network communications. Such networks are now known as **closed systems**.

The International Standards Organisation (ISO) realised that standards would need to be developed, for manufacturers to follow, so that inter-network communications (given the name **open systems interconnection**) could take place. In 1977, ISO undertook to produce a standard which defined open systems interconnection which, when followed by manufacturers, would mean that all networks could interconnect easily. The standard (**IS 7498**) has since been incorporated, only recently, into CCITT recommendation **X.200**.

4

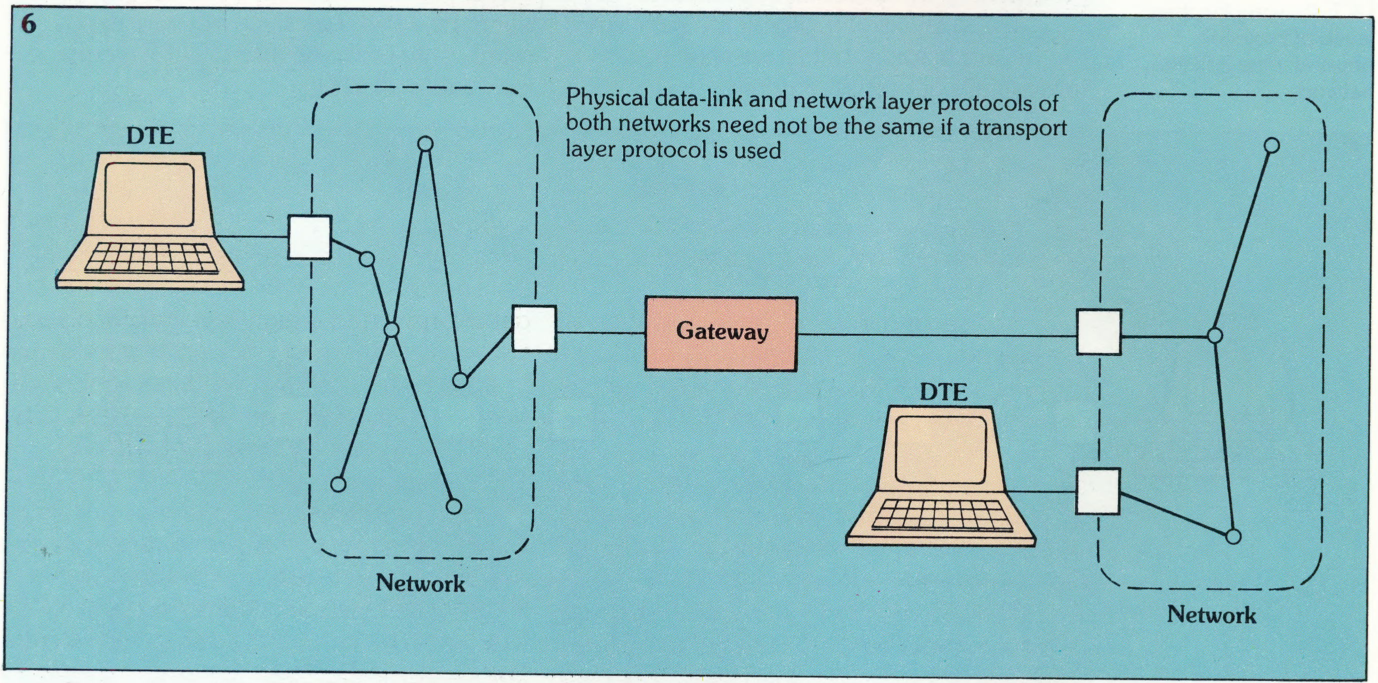


4. An example network of connected DTE and DCE showing the three levels of protocol: physical, data-link and network.



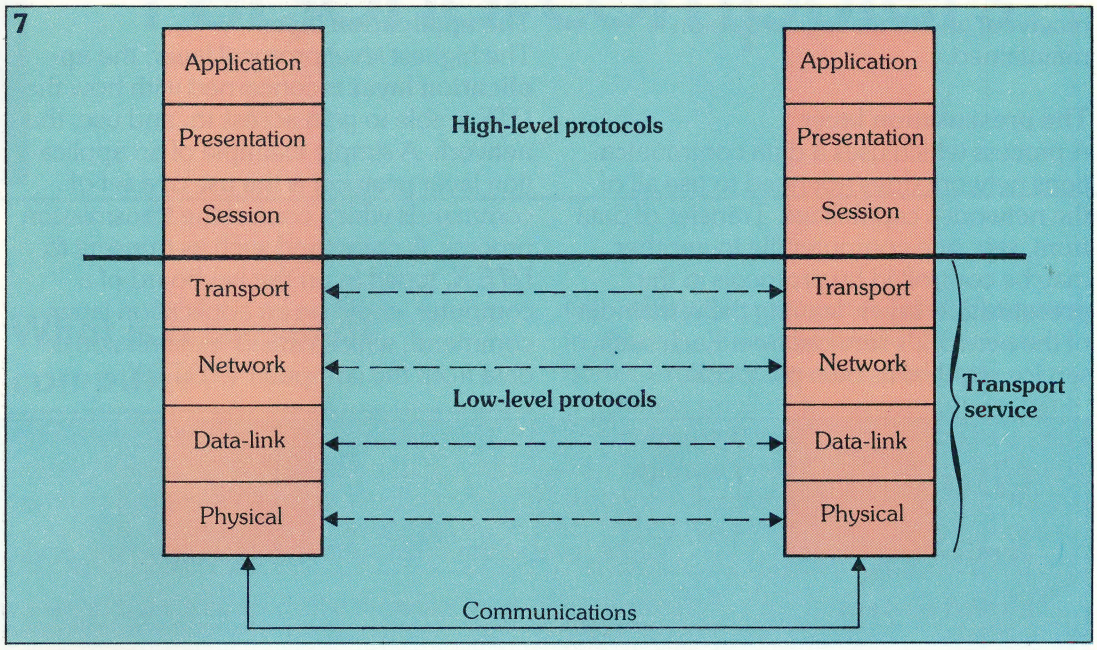
5. The open systems interconnection model for data transmission over the PSTN between two DTEs. Each protocol level is known as a layer.

6. Transport level protocols (level four) ensure that data transmitted from a DTE in one network is correctly received by a DTE in another network.


6
DTE
Physical data-link and network layer protocols of both networks need not be the same if a transport layer protocol is used
Gateway
DTE
Network
Network

Recommendation X.200 specifies a total of seven possible levels of protocol. The first three correspond to the electrical, data and network levels we have just seen, the other four we shall discuss shortly.

A slightly different terminology is used with reference to the open systems interconnection model of IS 7498 and X.200: the protocol levels are referred to as **layers**, i.e. the physical layer, the data-link layer etc. – except when referred to by number. They are then referred to by level, i.e. level one (the physical layer), level two (the data-link layer) etc. *Figure 5* shows a possible format of the model, connecting two DTEs, via modems and PSTN. Levels four to seven – the transport, session, presentation and application layers – are



7. In the open systems interconnect model, the transport service protocols – the transport, network, data-link and physical layers – are concerned with data manipulation.

not specified by this model. In more complex data communications models these higher level layers are important and therefore must be specified to allow open systems interconnection.

The protocol layers of the open systems interconnection model

As we've seen in the DTE/modem/PSTN example, the first level of a data communications system, the physical layer, is concerned with the electrical and hardware details of an interface.

The second level, or data-link layer, is concerned with the transmission of data in a meaningful manner – how the data is coded and decoded for transmission and reception.

The network layer, or third level, protocols control the data access to and through the network from one DTE access point (say, an exchange) to another.

The transport layer

Fourth level protocols, the **transport layer**, define the ways in which networks (each with a host computer) may be interconnected. *Figure 6* shows two networks interfaced by a **gateway**. Each network has its own three lower protocol layers, which need not necessarily be the same. Transport layer protocols ensure that the data transmitted from a DTE in one network is correctly received by the DTE in the other network.

The transport layer, like the three lower layers, is concerned with the *manipulation*, i.e. transport, of data between DTE. For this reason the first four levels of the open systems interconnection models are collectively called **transport service** protocols. *Figure 7* shows how the transport service layers correspond.

Levels of protocol above the fourth level, on the other hand, are often referred to as **high-level** protocols or layers, and are concerned with the *meaning* of the data transported – high-level protocols are said to *use* the transport service protocols. In effect, the high-level protocols do not need to be concerned with how the data is actually transmitted between DTE.

The session layer

Session layer protocols are concerned with the management of resources within the network. The network operating system software, for example, is governed by protocols of this level.

When two DTE are connected in a data communications link, they are said to be in a **logical session**. This may or may not be a single, permanent, physical connection but may be formed by a number of connections each with the function of transmitting a small amount of data. It is the responsibility of session layer protocols to ensure that the data is correctly received, and that a logical session is maintained.

The presentation layer

A process which uses a data communications network does not need to use all of the network's capabilities. Transfer of data from, say, one computer file to another may be controlled by protocols in the **presentation layer**, leaving the remainder of the network's services free. Each such service requires its own protocols.

The application layer

The highest level protocol layer, the **application layer** is concerned with how the DTE is able to gain access to, and use, the network. A simple example of an application layer protocol is the use of a set of commands which control the transmission process. A command such as *transmit to DTE X*, typed in on the keyboard of a computer, could be an application layer command, which causes transmission of data from the computer to DTE X.

Glossary

application layer	highest level of the open systems interconnection model of protocols
closed system	a communications network which cannot interconnect with another
data-link layer, (data layer, link layer)	second level of the open systems interconnection model of protocols
gateway	an interface between two networks
interface	where two systems or subsystems meet. An interface must converge the communications protocol followed by both systems
layer	a level of protocol of the open systems interconnection model
logical session	a communications link between two DTEs. The link may or may not be formed by a permanent connection for the duration of the exchange, as long as it appears to be by both DTE
network layer	third level of the open systems interconnection model
open systems interconnection	a communications protocol model which, if followed, allows interconnection of all networks
physical layer, electrical layer	the lowest level of the open systems interconnection model
presentation layer	sixth level of the open systems interconnection model
protocols	set of rules which provide the required discipline for communications over networks
session layer	fifth level of the open systems interconnection model
transport layer	fourth level of the open systems interconnection model
transport services protocols	protocols of the first four levels of the open systems interconnection model
X-series	CCITT recommendations concerning data communications over digital networks

14

COMPUTERS & SOCIETY

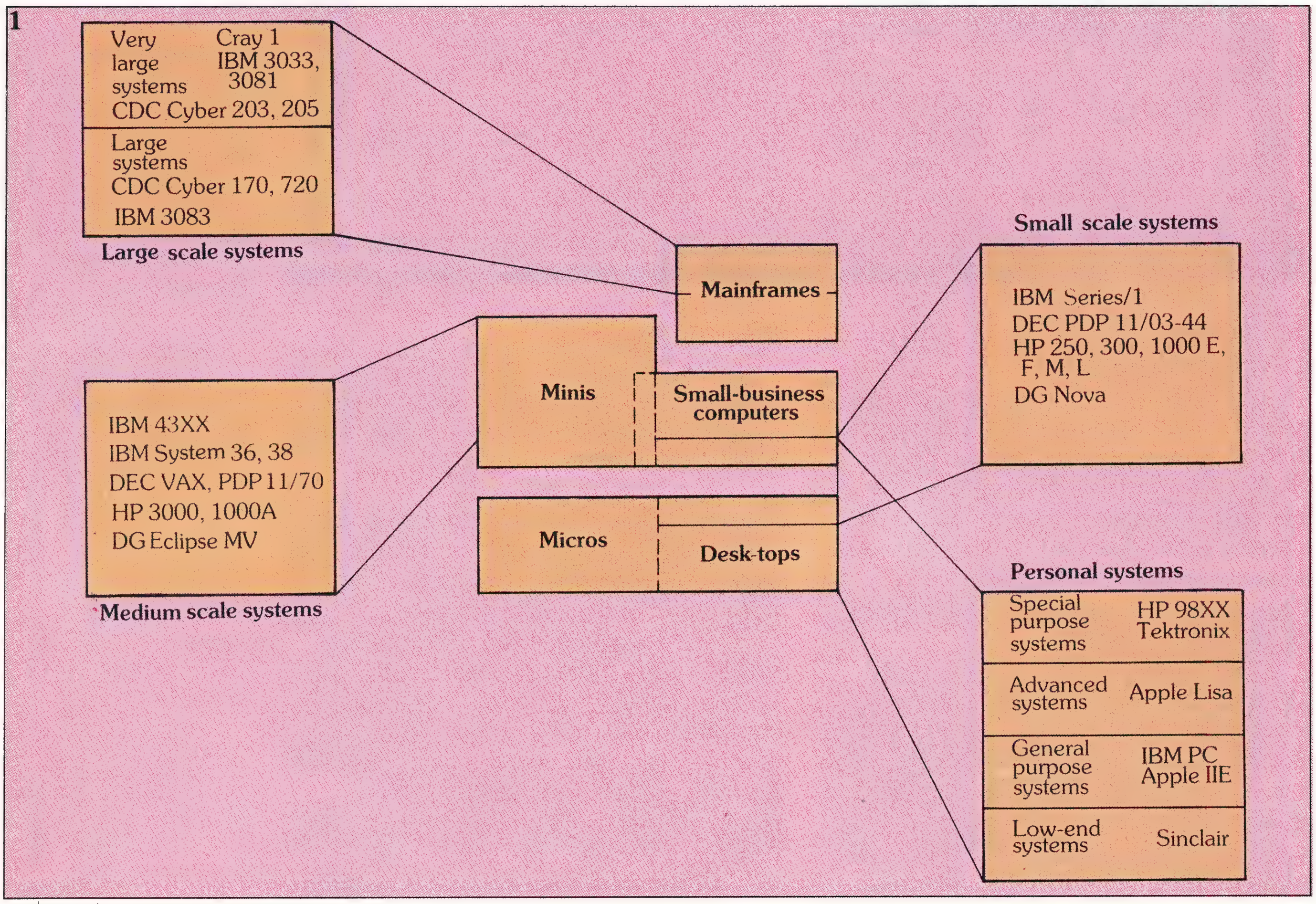
Personal computers

What is a personal computer?

An American market research company, International Data Corporation (IDC), has recently devised a new classification system for computers – it is based solely on *competition*.

Traditionally, computing systems were classified largely on machine **architecture** (i.e. the type, functions and interconnections of the various units and devices), however IDC's new method determines the appropriate class for machines by identifying the major products with which they compete. This is illustrated in *figure 1*, where the previous five categories – microcomputers, desk-top machines, small business computers, minicomputers and mainframes – are mapped into four new ones – personal systems, and small, medium and large scale systems.

This new classification has evolved because of the huge advances that have been made in the miniaturisation of computer technology: thus pushing microprocessor based systems to levels of performance hitherto attainable only by minicomputers. Another factor that had made classification difficult is the trend towards standard software: vendors of all classes of



1. Under International Data Corporation's new classification system, the previous five categories are mapped into four new ones, based on the products with which they compete.

Left: this Apricot xi features a 5 or 10 Mbyte 3½″ Winchester disk. The heads on the disk drive are moved according to an optimum positioning algorithm, reducing access times to an average of 85 ms. (Photo: ACT).

computer now offer the same operating systems and languages.

The IDC category 'low-end personal systems' encompasses home computers, such as those manufactured by Sinclair Electronics and Texas Instruments, for example. Just above 'low-end personal systems' is the category 'general purpose personal systems', which includes machines like the Apple IIE and the IBM PC (personal computer).

Although the IBM PC is the more powerful machine, both IBM and Apple are competing for the same section of the market – business and personal users. You would not expect to find a Sinclair ZX81 in an office for word processing, and neither would you expect to see an IBM PC in the home running Pacman (although these things do happen!).

For the purpose of this article, we shall consider personal computers to be those microcomputers found in business or professional usage.

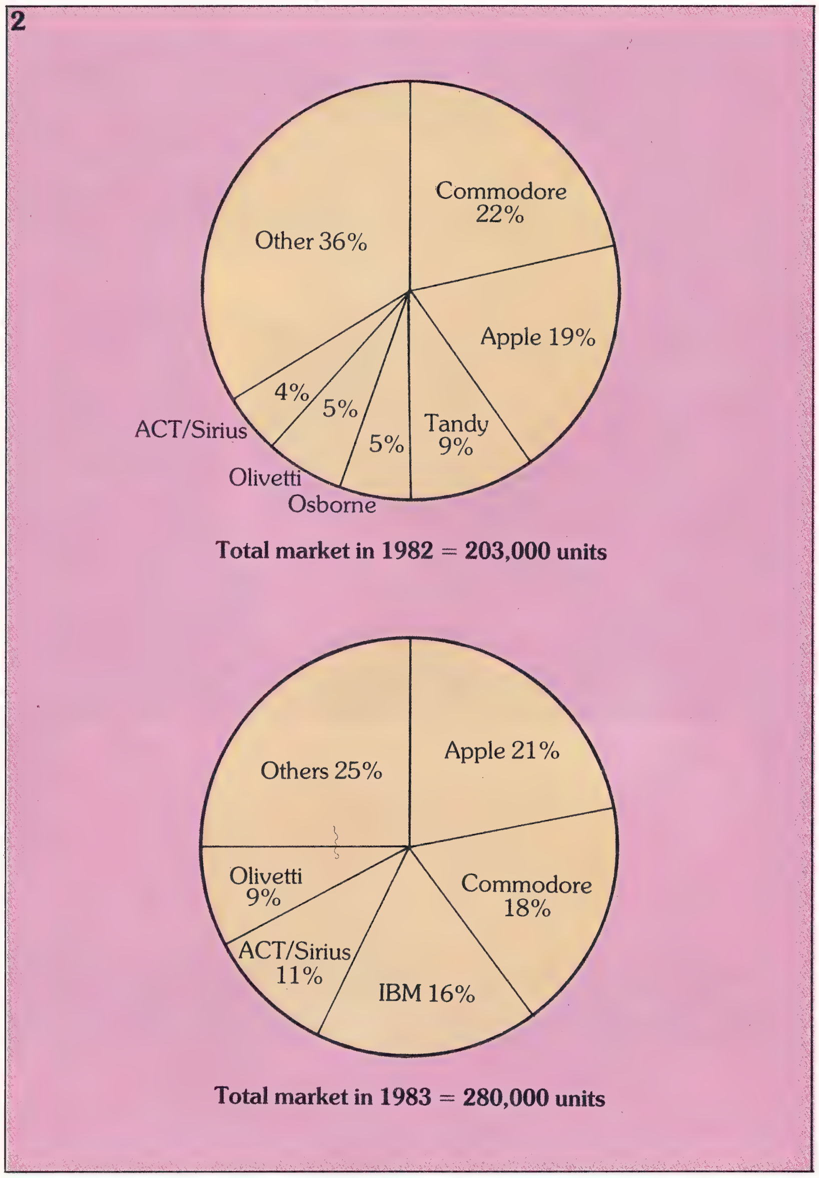
The evolution of the personal computer

The history of the microcomputer has been closely associated with two factors: reductions in size and increases in speed.

The reduction in size of computing systems is the more apparent trend. The vast mechanical calculators of the early computing days, for example, have been gradually replaced over the past couple of decades by desk-top machines, then machines that could be carried around in briefcases, and then pockets.

Computers have evolved from large

2



2. Pie charts illustrating how the personal computer market in W. Europe was split between the major manufacturers in 1982 and 1983.

mainframes, through minicomputers, and finally to microcomputers which are now small enough and sufficiently cheap for the individual to buy from high street retailers.

The arrival of the transistor brought about the first remarkable reduction in computer size and no sooner were the first transistors being manufactured than scientists were looking at ways of miniaturising *them*. Initially, about one cubic centimetre in size, they were soon reduced to the size of a pinhead. The miniaturisation of computer components was therefore fairly quickly achieved.

Transistor engineers then designed entire logic units – complete electronic circuits consisting of 20 to 100 components – fabricated on a chip of silicon about one centimetre square. Miniaturisation continued with large scale integration (LSI) and now very large scale integration (VLSI) techniques which enable thousands of individual units to be fabricated onto a semiconductor slice. And this process of miniaturisation is continuing.

The mainframe and minicomputer, of course, are not obsolete – but it is the microcomputer that has eased its way into our offices and homes in huge numbers.

Microprocessors

The microcomputer is a computer that uses a microprocessor to perform the same functions as the CPU in a mainframe or minicomputer. The first general purpose microprocessor was a 4-bit device, the Intel 4004, which was launched in 1971. The 4-bit architecture, however, proved too restricted for the microcomputer – it was too slow and cumbersome. Even its successor, the 8-bit Intel 8008 (developed in the early 1970s) required supporting integrated circuits to form an entire CPU.

Succeeding microprocessors included chips such as Intel's 8080, Motorola's 6800, Zilog's Z80 and the 6502 family produced by several manufacturers including MOSTEK. These needed fewer support circuits.

The evolution of faster and more powerful devices continues, until now we see 16 and 32-bit machines becoming the centre of attention as the microcomputer begins to claim much of the ground until now held by the minicomputer. The IBM PC, Apple II, III and Lisa (1 and 2), the ACT Apricot and Sirius, and the Wang Professional are examples of 16-bit office micros that have succeeded.

You can see from *figure 2* that, in Western Europe, this part of the market grew by 37% between 1982 and 1983.

32-bit processors

National Semiconductor and Texas Instruments, two of the world's largest semiconductor manufacturers, recently reached an agreement to jointly expand and develop National Semiconductor's 16000 family of 32-bit microprocessors, renamed the Series 32000. Hewlett-Packard has already built a 32-bit chip (with 450,000 transistors on it) which is incorporated in their 9000 range.

Motorola and Intel are also working on 32-bit chips and Zilog will be launching its 32-bit Z8000 early in 1985. Zilog's 32-bit chip will also be available from Nippon Electric Company (NEC). In addition, NCR and Honeywell are planning to use a 32-bit microprocessor in their new systems. A British company, Hytec Microsystems of Oxford, aims to base a 32-bit microcomputer series around this NCR/Honeywell chip.

Sinclair Research uses a Motorola 68008 chip, a 32-bit processor with an 8-bit data bus, in its fourth computer – the Sinclair QL.

The future of the microprocessor

Paradoxically, although microprocessor speeds have increased dramatically, speed continues to be one of the limitations on microprocessor performance; an electrical signal can only travel at about 150 cm per nanosecond (one billionth of a second). This can be partly overcome by packing the units on a chip more closely together, so that the electrons do not have to travel so far to deliver their message. One disadvantage of this approach, though, is that the closer the units are packed, the hotter they become.

Present research is considering ways of cooling systems down to absolute zero (minus 173°C). At this temperature, two phenomena occur: **superconductivity** and **tunnelling**.

Superconductivity

Superconducting material has no resistance to the flow of electrons. Therefore very little energy is required to initiate current flow and little or no heat is generated during electron movement.

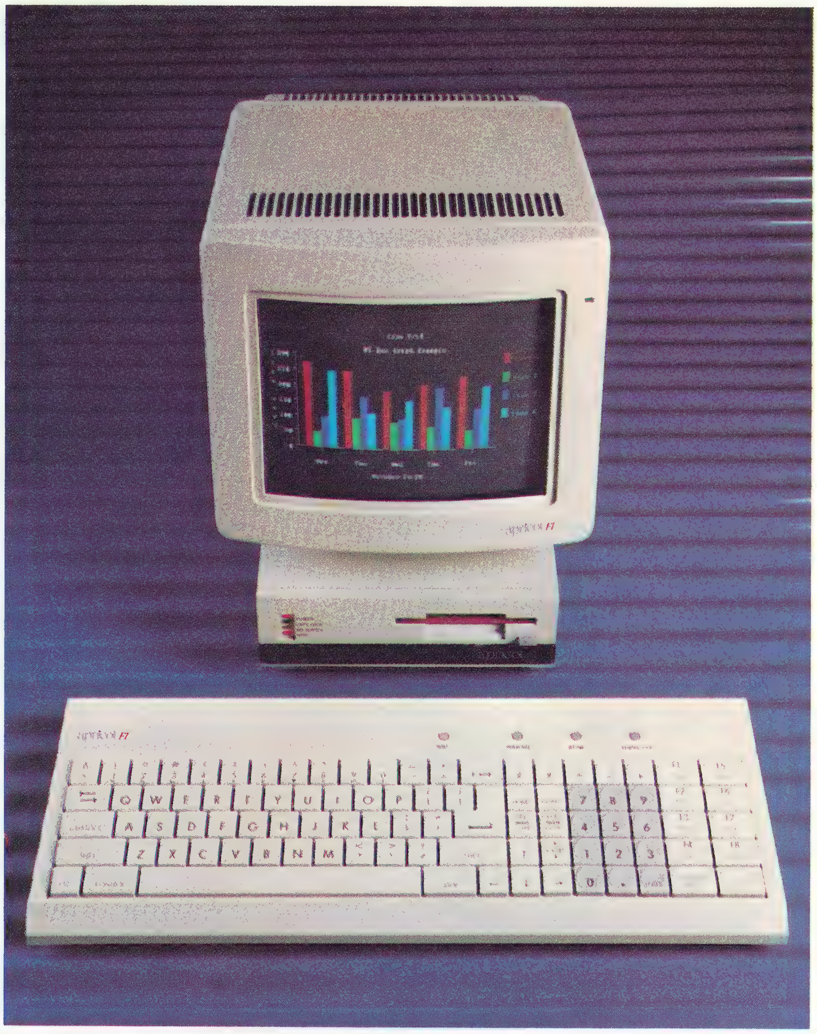
In an ordinary metal conductor, current flow occurs because electrons collide with the atoms of the metal conductor. In a superconductor, on the other hand, the current comprises paired electrons which move in such a highly co-ordinated manner through the metallic lattice that they never collide with the metal atoms.

Tunnelling

The phenomenon of tunnelling was discovered by Brian Josephson in 1962 – he was awarded the Nobel Prize for physics as a result. He found that at very low temperatures, electrons are able to tunnel through many electric insulating barriers. In fact, a moving electron may pass a barrier even though it lacks the energy to surmount that barrier. Considered impossible by classical physicists, this phenomenon is not inconsistent with modern quantum physics.

As a result of this breakthrough, the **Josephson junction** was developed. It exploits the fact that an applied magnetic field can suppress superconductivity. The presence or absence of a magnetic field can thus be used to represent a binary 0 or 1. It is the fastest switch available, capable of altering state in six picoseconds (six millionths of a millionth of a second).

Josephson junctions can also be used in memories to store information. Requiring power of only a few microwatts, so that a few million junctions would require only



Above: the Apricot F1 comes complete with a word processing package (SuperWriter), and a spreadsheeet, accounting package (SuperCalc). The colour monitor has a resolution of 640 × 256 pixels and any 4 colours from a choice of 16 can be displayed simultaneously. (Photo: ACT).

a few watts of power, a large computer using them would only need be a few centimetres square (but would need to be immersed in a bath of liquid helium).

The commercial exploitation of Josephson junctions, however, has presently come to a halt. After more than 15 years of research effort, IBM (the most influential exponent of Josephson technology) has wound down work on these high speed microcircuits. Costs of manufacture and packaging, together with the development of new materials, and new logic and memory circuits have been estimated at over $100 million.

Although the Japanese and other American companies, Sperry Univac and Honeywell, are considering Josephson technology and superconductivity, it looks as if this avenue of research for extremely fast computers has been closed.

Below: this IBM PC-AT is built around an Intel 80286 processor. Its proponents claim it to be 2-3 times faster than the Intel 8088 chip. (Photo: IBM).

The migration from data processing

About 20 years ago, computers started to take on a new role. From heavy duty calculators they now took on the more challenging role of information handlers, offering as they did, huge storage areas for vast quantities of data. As their main memory banks expanded, computers began to shrink, their processing speeds began to increase and their price fell dramatically.

The main motivating forces behind these developments were the American military and the space race. At first, computers were used to ensure that American defences were kept at a suitable state of readiness; then computers began to aid in the drawing up of defence strategies. Finally they were fitted into military vehicles providing on-the-spot information.

Computing therefore received a vast influx of funds from the U.S. Army. Many of the products generated by the Army reached the civilian marketplace in repackaged forms. Both these factors encouraged efforts to reduce the size of these machines.

The space race also provided a useful impetus to the development of American hardware. In 1957, when the Russians were launching satellites and dogs into space, the Americans were caught napping. The U.S. attempted to launch a series of rockets with their own technology and failed; the Russians remained ahead of the Americans for almost a decade.

The American response was to channel massive funds into attempts to achieve one goal: to land a man on the moon – first. This effort led to considerable advances especially in the area of miniaturisation as the Americans had to reduce the size of their machines to develop bigger booster rockets.

The third motivating force, commerce, led Thomas J Watson to establish one of the world's largest and most successful companies: IBM. Computers therefore became one of the capitalist world's expanding industries. Although, at first, the large organisations and government institutions involved were barely noticed by the general public, the arrival of the transistor and falling prices dragged the computer into the small business arena.

Successful companies ploughed profits back into research and development leading to more technological advances – and more profits. As bigger and bigger memories were squeezed into smaller and smaller spaces, new applications were discovered. The banks were among the first to discover data processing, with machine readable script at the bottom of cheques. Payroll systems and airline and hotel bookings soon followed.

At about the same time as Neil Armstrong was walking on the moon, the infiltration of computer technology into large business and government was considerable. Since then, they have become accepted in businesses everywhere.

Personal computers today

The personal computer has been able to invade the office through two routes: first, the installed base of electronic equipment in the office was already so high that to add a further item that linked them all together, thus magnifying their effectiveness, was quite cheap. Secondly, since the beginning of this decade, microcomputers for office applications had become more widely available through high street shops, as well as from an increasing number of distributors. Increasing demand plus increasing availability provided the spur to the widespread use of computers in the office.

The electronic machines that had been used in offices for some decades, then, were the precursors of the integral office of the future. (See *Computers &*

Below: the Apricot Portable is based on an Intel 8086 chip. The LCD flat panel display has a resolution of 640 × 200 pixels in graphics mode. It is truly portable – weighing only about 13 lbs. (Photo: ACT).

Society 3 for more details.)

Microcomputers designed for professional use, in either the office or industry, fall into two major categories: the **desk-top personal computers** and the more powerful bus-based desk-top computer. The term 'desk-top' was first coined by Hewlett-Packard to describe self-contained units that were intended to be user friendly. They were designed to talk to the user as soon as the unit was switched on. This is because the high level language normally used, BASIC, is present in read only memory and hence the machine does not need to load a machine code routine (boot) before programming can take place. The Commodore Pet is one example of a machine that used this approach and others soon followed.

Most desk-top units also incorporated cassette units for storing programs and data files. However, the falling price of disk drives and increasing program size opened the way for integral disk drives, in disk/tape subsystems.

The original desk-top machines used black and white or monochrome screens; green or amber screens came later. Apple Computer, though, was the first to use colour when its Apple II arrived on the market. Apple rapidly became the world's primary supplier of personal computers, a position that it still holds today.

The principal advantage of bus-based systems is flexibility: a system can be configured to the user's needs and can be easily expanded. Among the popular buses in use is the S-100, which is standard IEEE-696 of the Institute of Electrical and Electronic Engineers.

Now that extra memory units are available for non-standard bus based systems, the advantages of bus machines are not as significant as they were. Desk-top computers can now support larger amounts of storage in main memory, as well as on larger capacity disk drives.

The IBM PC heads the league of 16-bit machines. Launched in America in August 1981, it has rapidly become the industry standard. In comparison with many other personal computers, the machine itself is low powered and unspectacular, but the IBM label and the mass of software packages available for it have led to its widespread use. It is sold through roughly 130 independent dealers in the U.K. and costs less than £2,000.

Launched early in 1984, the Apple Macintosh is set to rival IBM's dominance in this area. It is based on the technology used in Apple's Lisa and costs £1,500.

The Macintosh is based on the Mototola 68000 chip, which is steadily increasing in popularity. The chip most often used in personal computers, however, is still the 8086/8080, which powers the IBM PC and most other 16-bit machines.

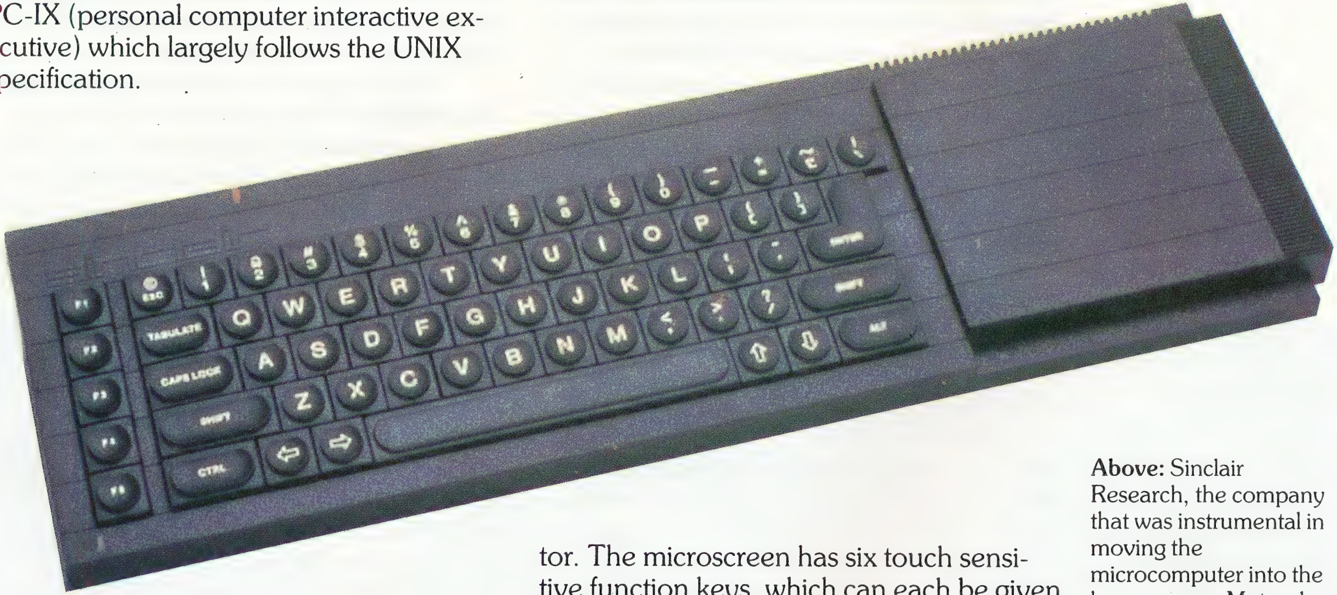
Although the choice of chips is fairly standardised, the old problem of 'which operating system?' still remains. For Intel's 8086 series, the MS-DOS operating system has moved ahead of CP/M-86 as the favoured operating system. This is, of course, partly due to the huge success of the IBM PC, which uses a slightly different version of MS-DOS called PC-DOS.

MSX personal computers

The **MSX specification** was first introduced by Microsoft in June 1983. The aim of the operating system is to generate low cost software which is portable across a range of manufacturers' machines. MSX defines the chip set, language, peripheral connections and the ROM cartridge format for a home computer. There is also a standard disk operating system, MSX-DOS, which shares the same disk format as MS-DOS. This allows data to be exchanged across 8 and 16-bit microcomputer systems.

A number of major consumer electronics manufacturers have announced that they intend to market and support MSX specification machines in the U.K. They include: Canon, Hitachi, JVC, Mitsubishi, Sanyo, Sony, Teleton and Toshiba. You'll notice that all of these are Japanese companies – it is they who have adopted MSX with the greatest enthusiasm. The MSX project provides one step towards standardisation of the microcomputer industry and the endorsement of MSX by so many major manufacturers should help the development of a broad base of software, benefiting both customer and software producer alike. However, as with the adoption of any standard, this might slow down the development of better systems.

The situation with the 68000 chip and operating systems, though, is still a little unclear. Apart from Apple's operating system running on the Lisa and the Macintosh, the only other user of the 68000 is Bell Laboratories' UNIX. Although there is only a relatively small amount of software available for it at present, this situation should change quite soon as enormous interest has been generated in this operating system. Early in 1984, IBM announced PC-IX (personal computer interactive executive) which largely follows the UNIX specification.



Above: Sinclair Research, the company that was instrumental in moving the microcomputer into the home, uses a Motorola 68008 chip in this Sinclair QL. (Photo: Sinclair Research).

Probably the most important new market entry is the new IBM PC which is faster and more powerful than other personal computers. The PC-AT (for 'advanced technology') is built around an Intel 80286 processor – this is claimed to be two to three times faster than the Intel 8088 chip used by IBM's three year old family of personal computers.

A top of the range PC-AT can address up to 3 Mbytes of main memory – five times the amount stored by ordinary PCs. This enormous memory enables this microcomputer to be used as a viable multi-user multi-tasking computer. This new machine also features a 1.2 Mbyte floppy disk drive and can support one or two 20 Mbyte hard disk drives. This means that the PC-AT can support twice the storage of the PC-XT, which can only store 10 Mbyte on a hard disk drive.

IBM also offers: a local area network which provides users with an inexpensive method of connecting personal computers to exchange data files; a new version of the IBM PC operating system; and a version of the UNIX operating system called PC-XENIX which is based on the XENIX operating system from Microsoft.

Another well known personal computer is ACT's Apricot, launched in September 1983. It is based on the 8086 chip, and features mouse control and twin 3.5 inch Sony floppy disk drives. On the keyboard there is an innovative LCD microscreen, which acts as a clock, calendar and calculator. The microscreen has six touch sensitive function keys, which can each be given a function unique to the program being run. It also enables the Apricot to be used as a portable computer, without a monitor. The Apricot costs £1,500.

The HP150 from Hewlett-Packard is also a 16-bit personal computer, but it uses a **touch screen**. The screen has an array of infrared scanners around the bottom of the display and along one side. Opposite each light emitting diode is a phototransistor, and as a finger touches the screen it blocks the light detected by one or more of the phototransistors. With this machine (at £2,900), Hewlett-Packard is aiming at the non-technical user.

PC markets

According to a survey by Sentry Database Publishing of Hudson, Massachusetts, sales of personal computers in North America in 1982 jumped by nearly 70%. In addition, the system integrators (those companies that add value to the price of computer equipment either by writing new software or by re-configuring the hardware) also expanded business in peripherals and application software packages.

The survey included 70% of America's systems houses and system integrators, who between them sold 171,000 computers last year to business and professional users. This mountain of hardware was accompanied by hundreds of thousands of software programs covering every conceivable business use.

Not all the results of the survey were consistent with a single trend, but there were a number of discernible directions. For example, for many users personal computers are not simple appliances that can be plugged in and put straight to work. To take advantage of the available technology, many users had to hire systems houses, or system integrators, to assemble and configure or program personal computer systems.

The availability of inexpensive computers has opened up new opportunities to value added remarketers in fields that have not been automated before. In particular, more and more software applications are being run on personal microcomputers for agriculture, food service, finance, publishing and the travel industries.

In Western Europe, a total of 203,000 personal computers were sold in 1982 (*figure 2*); this is expected to rise to over one and a half million units by 1989 (*figure 3*).

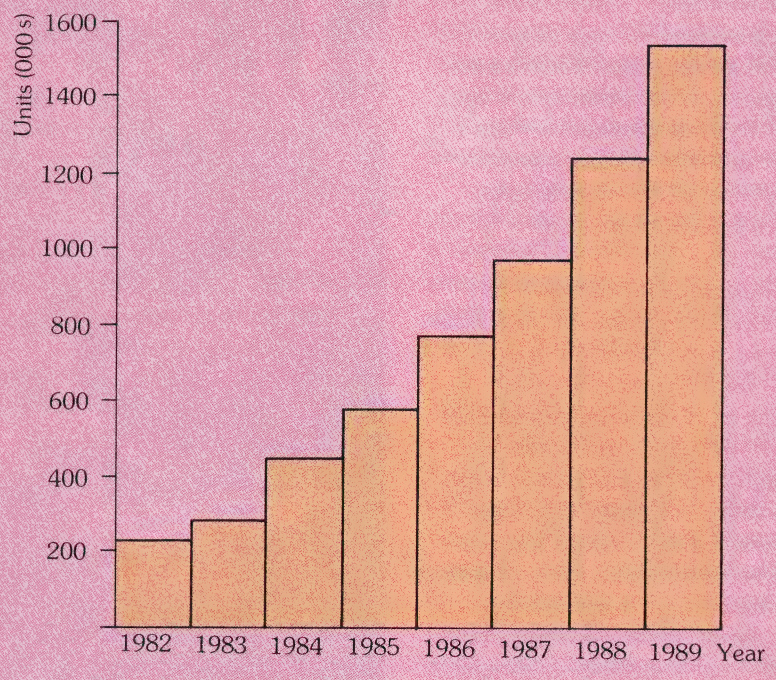
Personal home computers

For the first time, home computer manufacturers, such as Commodore and Sinclair, have overtaken the major mainframe and minicomputer companies in total sales.

The latest U.K. turnover figures, as reported in the BIS-Peddar 1983 survey, have even surprised the makers of these budget priced microcomputers. IBM still heads the turnover league, but Commodore is now ahead of minicomputer specialist Digital Equipment (DEC), and is now catching up with ICL. The survey shows that in the U.K. market, regular growth rates of 300% to 400% per year have been achieved. The U.K. population will have bought three million home computers by the end of 1984, with most of them being bought in 1983.

3. Bar chart showing the predicted increases in sales for personal computers in W. Europe to 1989.

3



Software

The software, of course, is just as important as the hardware it runs on. In some machines, the program is integrated into ROM. Some pre-programmed software is contained in special modules ready to be inserted into the system. Software contained in this type of permanent plug-in storage is known as **firmware**. This gives very fast access to frequently used programs. In general, however, the program is read, or loaded, into RAM from external devices such as magnetic disks or tape.

The increasing use of personal computers has accentuated the need for compatible software, i.e. software that can be used in the same form on different machines. The increasing number of inexperienced programmers has created an urgent need for packaged programs supplied directly by software manufacturers.

Examples of these kind of packages include the word processing package **Wordstar**, accounting software such as **VisiCalc**, and software for mailing lists such as **Mail-Merge**. These packages can run on any system supporting the CP/M operating system.

The choice of operating system is becoming more complex. CP/M is the dominating system for 8-bit machines, however for 16-bit machines there is considerable competition between CP/M and a clutch of new operating systems including MS-DOS (from Microsoft) and products like UNIX and PICK, which have migrated from minicomputers and mainframes.

The most obvious function of the operating system is to enable communication between the operator and the computer. At the very least, the system must provide the user with some way of indicating which program is to be run next. Indeed, most systems provide more sophisticated methods of running programs. Version 2 of the MS-DOS operating system, for example, is almost a programming language in its own right. Surprisingly, there seems to be no universal agreement on the functions that should be included in an operating system and those that can safely be made the domain of higher level software.

How has the PC affected society?

The technology of today's personal computers and the functions of the office of the future are starting to converge. The once separate areas of word and data processing, microcomputing and data communications are gradually being brought together. Most current office systems offer a range of facilities to the end user. Other forces are also working in this direction: there is an increased awareness of computers amongst end users, and hardware is also cheaper.

In addition to the falling prices, software is becoming more friendly and communications products are becoming more sophisticated. These are probably the two most important elements of tomorrow's office world. All these factors will help promote the shift from manual to electronic information processing. Microcomputers have made the integration of typical office functions possible. (See *Computers & Society 3*.)

Below: there are several different technologies in use in the design of touch input systems. This example, from Hewlett-Packard, uses two perpendicular infra-red beams. A touch is recognised when the beams are broken. Inverse video is used to display the point chosen, but the choice is not acted upon until the finger is withdrawn and the beams are again detected.
(Photo: Hewlett-Packard)

Right: this hardware forms the basis of the Apricot Point 32 local area network. The file server on the right has a 10 Mbyte Winchester disk and can support up to 32 individual personal computers for shared access on the network. Mass storage is provided by the 100-200 Mbyte Random Access tape drive on the left. (Photo: ACT).

Specialist uses

The ever increasing availability of cheap microcomputers has led to their increased use in a wide variety of applications. Some of these applications require only moderate computing power and a dedicated stand-alone microcomputer, adequate for monitoring and controlling small processes.

Large industrial plants distributed over several geographical locations, however, may require a distributed control approach, where a number of personal microcomputers are placed close to the plant and connected through a serial communications network. The personal computer as it stands today has a role to play in CAD and computer based management information systems, but as soon as 32-bit machines become generally available, personal computers will also be able to be used in CAD processes.

Financial accounting

The use of spreadsheet programs by microcomputers can alleviate some of the accountant's more boring tasks. Once a cash flow has been transferred to a micro-based spreadsheet program, an accountant only has to key in new data and the program will, on command, recalculate the complete cash flow statement.

Probably the best known financial program is **VisiCalc**, which was introduced in 1979. VisiCalc, however, is being superseded by a new package, **Lotus 1-2-3**. One view of the explosion of personal computing in business is that it has been largely due to these financial 'calc' programs. In a survey of desk-top computers earlier in 1984, the Financial Times wrote that 70% of IBM PCs had been bought solely to run Lotus 1-2-3. Perhaps the reason for this is that spreadsheet programs are very friendly – they have given non-technical people the ability to solve their own data processing problems.

On the whole, electronic spreadsheets are quite versatile and they support a range of applications. A typical spreadsheet design uses 63 columns by 254 rows, although the actual size varies from package to package.

Although spreadsheet programs are being used increasingly on minicomputers and mainframes, they do seem best suited to personal computers. Spreadsheets are interactive programs similar to word processing software in many ways, and a dedicated PC system provides the maximum response. Larger systems typically use software that is less versatile, with less agile scrolling functions. Timesharing can also degrade the performance, because there are delays while the host processor works on someone else's program.

The future for personal computers

The power of the stand-alone microcomputer is limited: they are fine for word processing and for electronic mail, but they can only handle financial plans of limited size and complexity. Another drawback is that the microprocessor's facilities for data management are limited to quite small file sizes. However, once connected to a network they become much more powerful tools.

What personal computing offers to business is processing power on everyone's desk. This will make access to data on the central mainframe more flexible and much easier to cope with.

A number of high level languages have been developed that run on both microcomputers and mainframes. These have two benefits: first, they allow programs developed on a micro to be run on a company's mainframe, hence reducing the time and cost of program development; second, libraries of programs can be accessed by managers on their own personal computers. This provides some standardisation of program quality and yet maintains the independence of managers at their micros.

Companies have large amounts of information stored on their central machines. This information is of greater benefit if a large number of people have access to it. Access to that information, through a micro, enables many individuals to analyse data and present results in the most convenient way (hard copy print-out, colour graphics, or even synthetic speech) at a cost that can be justified.

Technology

The personal computer of the future may turn out to be an **optical computer**. Pulses of light moving down an optical fibre are vastly more efficient than pulses of electrons moving down a copper cable. An optical switch known as a **Fabry-Perot interferometer** which, claim its supporters, is as fast as a Josephson junction could become the basis of computer technology in the future.

If the current developments in fibre optic networks rapidly become accepted, then during the 1990s optical computers may well replace those based on integrated circuits. The problem with fibre optic technology is that it is far too expensive at the moment for the personal computer market, which is extremely cost-conscious. As the techniques for implementation become better and cheaper, however, fibre optics will move into the mini/micro area.

There are many other technologies under development that will effect both the core of the computer and the peripheral devices that store and transmit information. For example, **laser based data storage** and **satellite** based transmission systems.

Bubble memory, another new development, lies (speed-wise) somewhere between semiconductor memory and magnetic disk memory. While it will not supersede semiconductor memory as the main processor memory, it will allow faster access to back-up storage, and thus indirectly increase processing speeds.

As computers become cheaper and more compact, they will be used in more and more applications. Microcomputers are increasingly being used to control industrial robots. However, as yet, relatively few contain any significant degree of machine intelligence.

The next major development in personal computer controlled robots will be the introduction of some form of intelligence. And with the increasing amount of interest being generated by expert systems this does not seem too far away.

The Japanese expect to have a prototype 'thinking' computer by 1990 as part of their 'fifth generation' project.

Perhaps the biggest development during the next few years will be the move to multi-user systems. These will be built upon multi-microprocessor systems using multi-user operating systems. Personal computers combined with some standard networking technology will be able to share expensive resources to a greater and greater extent.

These new systems will then pose a serious threat and an ultimately fatal challenge to traditional medium and small scale personal computer systems.

ELECTRICAL TECHNOLOGY

Transmission lines

So far we have studied all electrical systems on the assumption that they comprise a variety of circuit elements (in which various modifications of signals occur) which are interconnected by a network of wires. We have assumed that these wires possess ideal properties, so that the voltage at one end is identical to that at the other, and that the current entering the wire is identical to the current leaving it. This system model is completely satisfactory providing the currents and voltages are of a sufficiently low frequency.

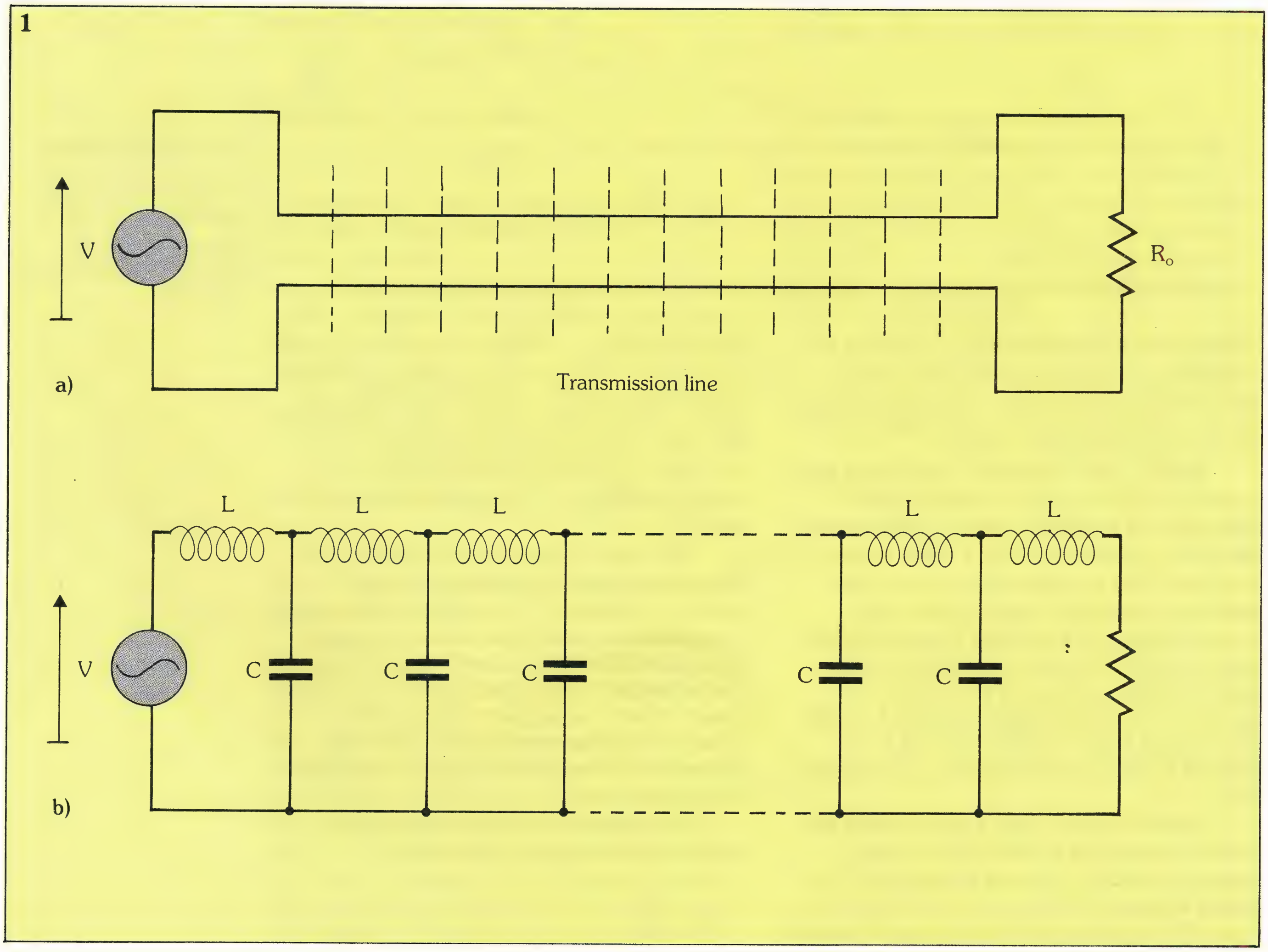
We'll look at this in a bit more detail. Any conductor used to transmit electrical or electromagnetic energy is defined as a **transmission line**. Most transmission lines do affect a circuit's action as their inherent capacitance makes them act as low pass filters. To be a little more precise, we can say that any circuit or transmission line of overall length about 10 cm can be analysed on the assumption that signals below 300 MHz will pass through it unaffected. Above this limit, the electrical properties of the transmission line must be accounted for.

As a very simple example, consider the circuit in *figure 1a* comprising a voltage generator connected by a transmission line to a load resistance R_o. This might represent the connection between a generating station and the consumer, or between a radio transmitter and the aerial, or even between one part of a computer and another. The transmission line may take the form of a pair of parallel wires, or a coaxial cable.

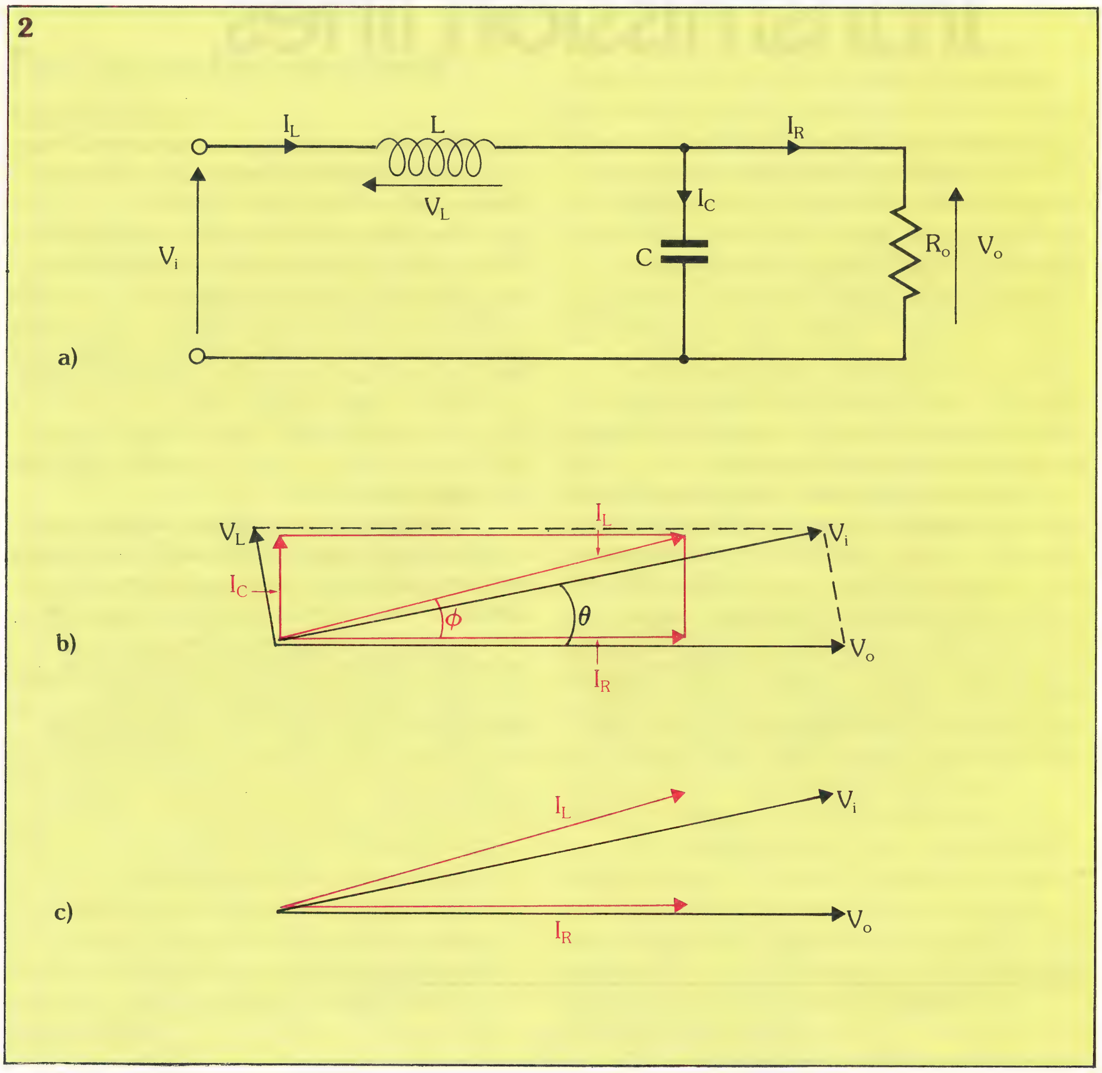
The transmission line's two conductors act like the two plates of a capacitor, and the air or other insulator between them forms the dielectric; each short length of the transmission line therefore represents a capacitance. Looking at the complete circuit and starting from the top of the voltage generator, if we trace through the resistor and then back along the

1. (a) A simple circuit comprising a voltage generator connected by a transmission line to a load resistance, R_o; **(b)** breaking the line in (a) into a number of short lengths produces this circuit model.

1



2



2. (a) Circuit diagram for a section of lossless transmission line with load resistor, R_o; **(b)** phasor diagram; **(c)** phasor diagram redrawn for clarity.

bottom wire to the generator, we find that this comprises a singe loop of wire which can be represented by an inductor. As a consequence, we may assume that each short length of wire may be *modelled* by an inductor.

Breaking the transmission line into a large number of short lengths (as shown by the dotted lines in *figure 1a*), may be modelled by the circuit in *figure 1b*. Here, L is the inductance and C the capacitance between wires made up of one short length of line. (For convenience we will consider this short length to be 1 metre.) You will notice that we have neglected the resistance of the wires: this is usually an acceptable approximation for most transmission lines. We shall call such lines **lossless** as the power lost in them is negligibly small.

To find out how such a transmission line works, consider the section with the load resistor (of value R_o) shown in *figure 2a*. The phasor diagram for.this small circuit is given in *figure 2b*. Beginning with the voltage V_o across the load resistor, we find the current I_R through R_o in phase with V_o, and the current I_C through C leading V_o by 90°. Adding these two by the phasor parallelogram rule we get I_L, the current through L. Now the voltage V_L across L leads the current I_L by 90° as shown. Finally to find the input voltage, V_i, we add the phasors V_o and V_L.

The input voltage V_i and current I_L, and the output voltage V_o and current I_R are redrawn in *figure 2c*. We can see that the input voltage is almost exactly equal to the output voltage, and the input current is also almost exactly equal to the output current, but that the phases of both current and voltage have moved forward from the output to the input. Of course this depends on the relative sizes of the various phasors.

The magnitudes of the currents and voltage in this circuit are written as:

$$I_R = \frac{V_o}{R_o}$$

$$I_C = 2\pi fCV_o$$

The angle, ϕ, between I_R and I_L is thus given by:

$$\tan\phi = \frac{I_C}{I_R} = \frac{2\pi fCV_o}{V_o/R_o}$$
$$= 2\pi fCR_o$$

If C is small (which would be true if the transmission line was divided into very small sections), the magnitude of $I_L = I_R$. Again, from the phasor diagram we see that:

$$V_L = 2\pi fLI_L$$
$$= 2\pi fLI_R$$

and the angle θ between V_i and V_o is given by:

$$\tan\theta = \frac{V_L}{V_o}$$
$$= \frac{2\pi fLI_R}{R_o I_R}$$
$$= \frac{2\pi fL}{R_o}$$

Looking at *figure 2c* we see that the angles θ and ϕ are equal, so:

$$\frac{2\pi fL}{R_o} = 2\pi fCR_o$$

from which we can see that:

$$R_o = \sqrt{\frac{L}{C}}$$

This particular value of resistance is known as the **characteristic resistance** of the transmission line.

It is also interesting to note that since the voltage and current across the load are related by:

$$V_o = R_o I_R$$

and that:

$$|V_i| = |V_o| \text{ and } |I_L| = |I_R|$$

we find that:

$$V_i = R_o I_L$$

In other words, the input resistance of this small section of line given by V_i/I_L is equal to R_o. Consequently we can repeat the preceding argument for each section all the way back to the generator, since each section is effectively terminated by a resistor, R_o.

Therefore, if we take a transmission line terminated in its characteristic resistance and supply it with a sinusoidal voltage, the magnitude of the voltage at any point along the line will be exactly the same as at the generator. However, the phase of the voltage will lag that of the generator, progressively more and more, as the point of measurement moves further from the generator.

In this manner, the waveform thus appears to travel along the transmission line from the generator to the load. It is therefore called a **travelling wave**. If we were to concentrate our attention on the maximum value of the wave we would see that it moves along the line with a velocity of u ms^{-1}. This is known as the **velocity of propagation**, or phase velocity, and is given by:

$$u = \frac{1}{\sqrt{LC}}$$

This velocity is always less than, but usually fairly close to, the velocity of light (3×10^8 ms^{-1}). It is related to the velocity of light, c, by:

$$u = \frac{c}{\sqrt{\mu_r \epsilon_r}}$$

where μ_r is the relative permeability and ϵ_r is the relative permittivity of the medium surrounding the line.

As a consequence of this, if we supply energy to a correctly terminated line (one which has a load resistance equal to its characteristic resistance) then all the power will be delivered to the load resistance. However, it will arrive at the load delayed by a time:

$$t = \frac{l}{u}$$

where l is the length of the line.

As an example, consider a pair of conductors of length 10 cm that connect adjacent circuits in a computer. The velocity of propagation is 2×10^8 ms^{-1}, so the time delay will be 0.5 ns. You can easily see from this, that the speed of a computer is limited by the distance between adjacent circuits. □

11

MICROPROCESSORS

8-bit microprocessors

Applications

The 4-bit microprocessors we have discussed so far do not possess an **interrupt** capability – they have to initiate the time at which they receive inputs from outside. This means that timer subprograms need to be incorporated to determine *when* the microprocessor checks to see if external information is being sent on its input lines. This is a simple method, but not always effective – if an external signal is sent between checks, then that signal is missed.

An **interrupt driven** microprocessor, on the other hand, lets the external signals themselves tell the microprocessor when communication is needed. In this way, all inputs are received and the microprocessor does not waste time looking for inputs that are not there.

Only some 4-bit microprocessors are interrupt driven, but almost all 8 and 16-bit devices have this interrupt capability. In this chapter, an example application will be developed that depends almost entirely on the interrupt features of the TMS8080A microprocessor. This will provide an insight into the architecture and instruction set of a very popular 8-bit microprocessor.

The most obvious point in favour of the use of an 8-bit microprocessor is that it offers double the speed of handling arithmetic and communications operations, and handles twice as much information as a 4-bit device.

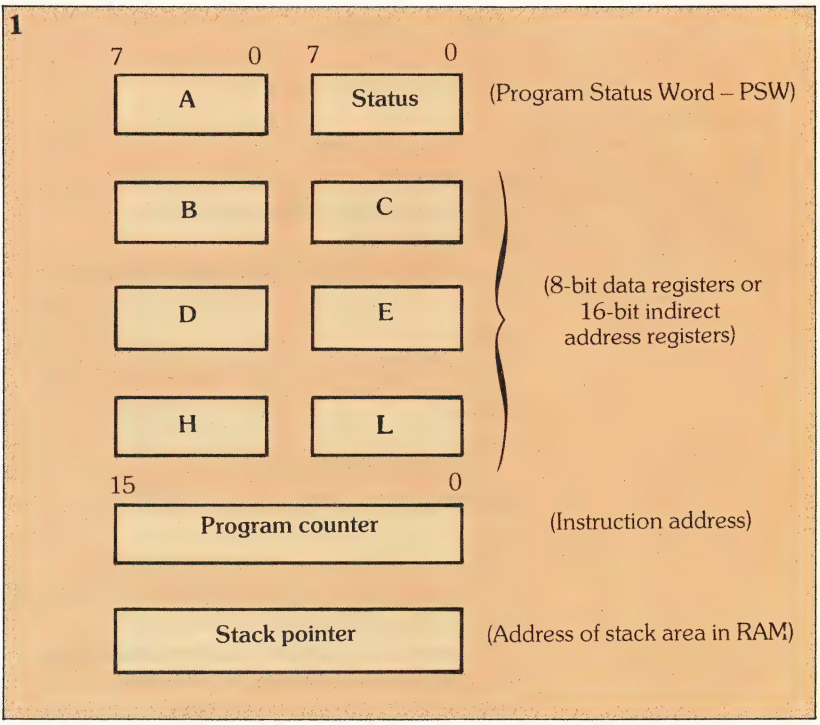
Although this would not be an advantage in simple control operations using single digits up to 16 (i.e. 4 bits), it does become important when dealing with the transmission of 8-bit ASCII characters or in arithmetic operations requiring accuracies of greater than one part in 16. Even in machines that need decimal arithmetic, 8-bit devices process two decimal digits at once – compared with the one-digit-at-a-time performance of 4-bit machines.

A typical 8-bit microprocessor

The TMS8080A microprocessor supports a relatively complete instruction set and a simple, effective interrupt structure. While its timing features are relatively straightforward, they are not as simple as other 8-bit or most 16-bit microprocessors.

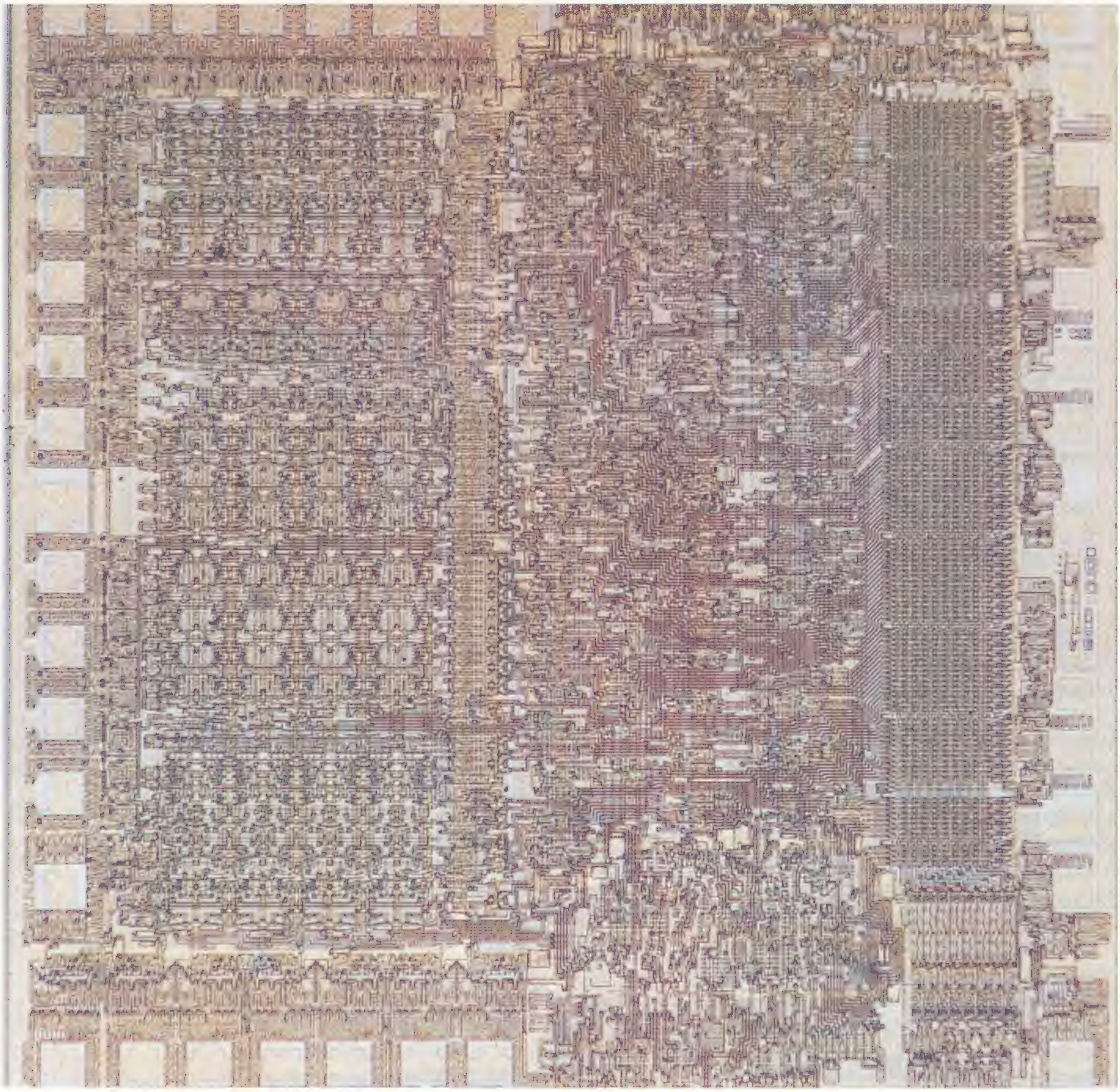
This microprocessor primarily depends on *register addressing* to locate data held in memory, and it provides seven internal general purpose 8-bit registers for high speed data manipulation.

1. The TMS8080A internal registers available to the programmer.


1
7
0
7
0
A
Status
(Program Status Word – PSW)
B
C
D
E
H
L
(8-bit data registers or 16-bit indirect address registers)
15
0
Program counter
(Instruction address)
Stack pointer
(Address of stack area in RAM)

Inside the '8080A

The registers available to the programmer are shown in *figure 1a*. Register A is the accumulator which is, of course, involved in all arithemtic and logic operations. The status register contains the zero, sign, carry, half-carry (used in BCD arithmetic with the DAA instruction) and the parity flip-flops.



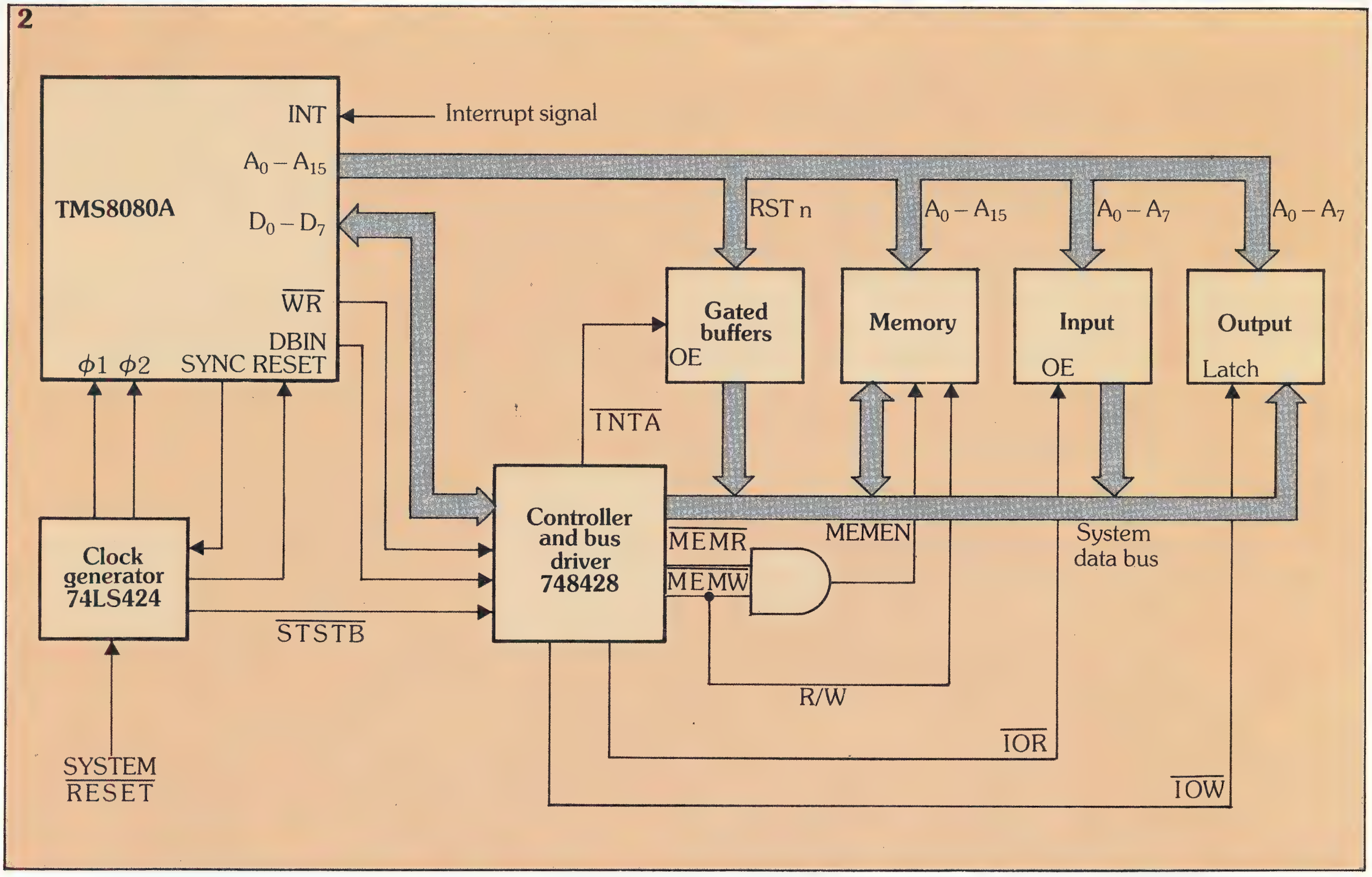
Above: photomicrograph of the 16-bit EF 68000 microprocessor. (Photo: Thomson-CSF).

The parity flip-flop is set to 1 when the number of ALU bits equal to 1 is even.

The interrupt flip-flop and interrupt-enable flip-flop monitor and control the microprocessor's interrupt signalling. Registers B, C, D, E, H and L are used as general purpose 8-bit data registers. They can also be used in pairs (B-C, D-E and H-L) to act as 16-bit data address registers (register indirect addressing mode). The program counter (PC) contains the instruction address.

A special set of memory locations, known as a **stack**, are set aside as operations or working registers. The name stack is used because information is stacked – one location on another. The register which keeps track of where information is or where it can be stored in the stack, is known as the **stack pointer** (SP). This contains the address of the currently available empty location in the stack.

Information is normally pushed down the stack, one location after the other; or it



2. TMS8080A system timing and control.

is pulled up from the stack one location after another. In some microprocessors, the stack of registers is located within the microprocessor; with the '8080A, the stack of registers has to be provided in external RAM storage.

Whenever a subroutine is called in the program, the contents of the program counter needed for the next instruction after the call are saved in the stack. A RETURN instruction fetches this value off the stack and sends it to the program counter to effect the return from the subroutine. There are also instructions that allow the programmer to push the microprocessor registers to, and pull them back from, the stack 16 bits at a time.

In order to use these stack operation features, an additional storage area must be set aside in the RAM data memory. The address of this additional storage area (last byte in the area) must be loaded into the stack pointer when the processor goes through its first initialisation procedures, in order to load the correct initial values into the stack.

Of course, the microprocessor has many registers and logic units that are not shown in *figure 1*. There must obviously be an instruction register, instruction decoder and arithmetic logic unit. The microprocessor user knows that these are present, but need only be concerned with the registers that are accessible through the device's instructions and operating features.

Timing and control

The timing and control signals that are sent to and received by the microprocessor are of special interest to the hardware designer. These features include the address and data signal timing, the interrupt signals, and the memory and input/output control signals. 65,536 memory locations can be addressed by the 16-bit address bus of the TMS8080A.

In addition, the TMS8080A can address 256 input and output devices. The same address bus is used for selecting these input or output devices, but only the least significant 8 bits of the address have any meaning in input/output operations.

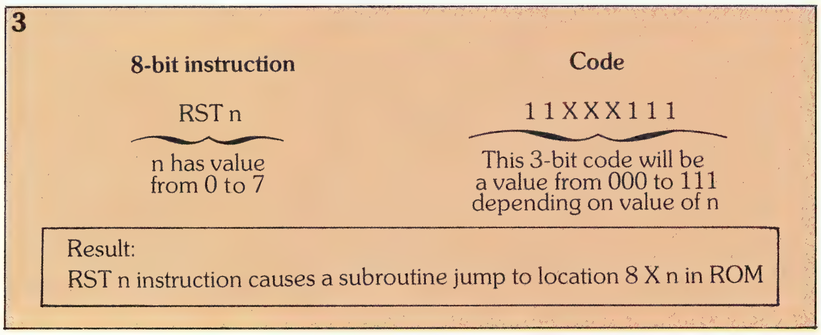
An operational status byte is provided

at the beginning of each operation so as to alert the external components. This status byte contains the information that a memory write, a memory read, an input read, an output write, and interrupt acknowledge, etc. are in progress.

These signals are not the enables themselves, but are used as *sources* to provide the memory enables that turn on the memory subsystem and the input/output subsystems. As a result, the necessary read/write signals can be generated.

Read/write signals

The simplest way of generating read/write control signals is to use two peripheral ICs designed especially for the '8080A: the SN74S428 system controller and bus driver; and the SN74LS424 clock generator.


3
8-bit instruction
Code
RST n
1 1 X X X 1 1 1
n has value from 0 to 7
This 3-bit code will be a value from 000 to 111 depending on value of n

Result:
RST n instruction causes a subroutine jump to location 8 X n in ROM

3. TMS8080A RESTART instruction code and operation.

The control structure is shown in *figure 2*. DBIN is the read timing pulse; $\overline{\mathrm{WR}}$ is the write timing pulse. When these signals are fed to the SN74S428 along with the operational status word on the data bus, the system controller provides the correct control signals of the proper duration at the correct time to interface the TMS8080A to external memory or to external I/O units.

For example, MEMEN (memory enable) and R/W for external memory are shown in *figure 2*. MEMEN is the NAND result of $\overline{\mathrm{MEMR}}$ (memory read) and $\overline{\mathrm{MEMW}}$ (memory write). R/W results directly from memory write, $\overline{\mathrm{MEMW}}$. Similarly, $\overline{\mathrm{IOR}}$ (input/output read) out of the controller is used directly to control information sent from the inputs to the microprocessor; and $\overline{\mathrm{IOW}}$ (input/output write) latches information coming from the microprocessor into an output unit. These signals are fairly straight-forward; the use of the $\overline{\mathrm{INTA}}$ (interrupt acknowledge) signal is not so obvious, though, and requires an examination of the way in which TMS8080A responds to an interrupt to fully understand its purpose.

Interrupt control

The '8080A will respond to an interrupt signal received on its INT pin when it has finished its current instruction operation, if the interrupt enable flip-flop has been set with the EI (enable interrupt) instruction.

The microprocessor responds by sending out an interrupt acknowledge signal and waits for a special subroutine jump instruction to be returned to it. This instruction, RESTART instruction (RST n), must be generated by hardware external to the '8080A. The $\overline{\mathrm{INTA}}$ signal provided by the controller is used to gate RST n onto the data bus, so that the microprocessor can receive it at the proper time. The format of the instruction is shown in *figure 3*.

The 3-bit code in the RST n instruction is generated by the hardware and causes a jump to the address in ROM formed by the 3-bit code followed by three zeros. So, if the 3-bit code is 000, a jump to location 0 in ROM occurs; if the 3-bit code is 001, a jump to location 8 occurs, and so on.

The programmer has to provide instructions at each of the addresses in ROM that start a subroutine to respond to a given type of interrupt, so that the subsystem or event that caused the interrupt can be serviced. The hardware designer must ensure that the INT signal is generated, and that the correct restart code for each interrupting subsystem is provided.

Once the subroutine has serviced the interrupt, the '8080A returns to whatever it was doing in the program at the time of the interrupt, provided the programmer ends his interrupt subroutine with a RET instruction. Prior to the interrupt, the programmer enables the interrupt system with an EI (enable interrupt) instruction. The interrupt system must be re-enabled after an interrupt somewhere in the interrupt subroutine with an EI instruction. Otherwise the processor ignores any INT signal until it does encounter an EI instruction.

For all interrupt procedures it is assumed that the hardware designer has provided a RAM storage area in data memory to save, in the register stack, the program information being executed before the interrupt occurred. Communications with the '8080A are established and controlled with the interrupt related instructions: EI (enable interrupt); RST (subroutine jump to interrupt procedure); and DI (disable interrupt) which locks out an interrupt. The rest of the '8080A instruction set provides the microprocessor's normal data manipulation and program control capabilities.

'8080A instruction set

The '8080A (like most microprocessors) offers a variety of data movement, logical, comparison and branch operations. Of these, the data movement instructions shown in *figure 4* are the simplest to use and understand.

The basic instruction is the MOV d,s where the source, s, and destination, d, is any one of the microprocessor registers A,

4

Data move instruction

MOV d,s s→d Move contents of s to d.
s and d can be A,B,C,D,E,H,L, or M with M being the memory location whose address is in the HL register pair

8-bit loads and stores of the accumulator

INDIRECT ADDRESSING MODE:

LDAX	rp	Load the accumulator with data in memory whose address is in the register pair indicated
STAX	rp	Store the accumulator contents in the memory at the address contained in the register pair

rp can be the BC, DE, or the HL register pair

DIRECT ADDRESSING MODE:

LDA	address	Load the accumulator with data from the memory at the address contained in the instruction
STA	address	Store the accumulator into the memory at the address contained in the instruction

Initialising registers and register pairs

8-BIT REGISTER INITIALISATION:

MVC	d,data	Send data from instruction into register d with d being A,B,C,D,E,H,L or M

16-BIT REGISTER INITIALISATION:

LXI	rp, value	Place 16-bit value from instruction into the register pair indicated rp can be B,D,H or SP

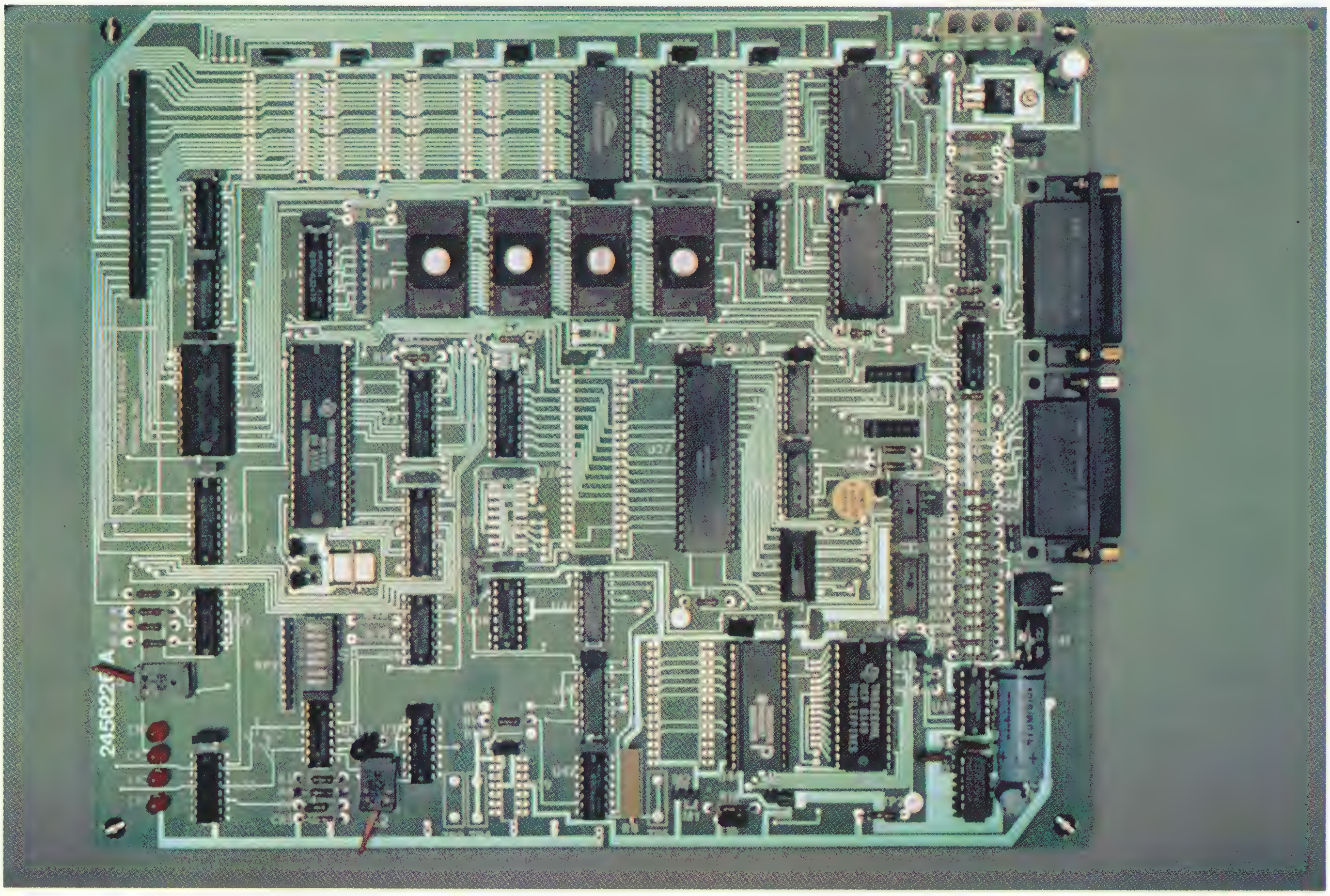
16-bit transfers

XCHG		Exchange contents of HL register pair with DE register pair
PUSH	rp	Push the register pair contents to the stack
POP	rp	Pop the register pair contents from the stack
SHLD	address	Store the contents of the HL register pair in memory starting at the address in the instruction
LHLD	address	Load the HL register pair with the 16 bits in memory addressed by the address in the instruction

Input output transfers

INP	address	Input data to the accumulator from the input device at the location specified by the address in the instruction
OUT	address	Output the accumulator data to the device at the location specified by the address in the instruction.

4. Data move instructions for the TMS8080A.



Above: vocal synthesis system on a plug-in printed circuit board. This operates by linear predictive coding principles.

B, C, D, E, H or L. s or d can also be the memory location specified by the address in the HL 16-bit register, in which case the s or d operand will be denoted M. The result of this operation is that the data in the source location is copied into the destination location.

There are also indirect addressing load and store operations that act on the accumulator through the LDAX rp and STAX rp instructions. rp (register pair) is D (for the DE pair) or B (for the BC pair). The direct addressing version of the load and store accumulator is the LDA address and STA address, respectively, where the address of the memory data location is contained in the 3-byte instruction.

The INP address and OUT address instructions either move data to the accumulator or from the accumulator to an input/output register specified by the address (which must represent a number between 0 and 255).

Other special purpose movement instructions are available to exchange the contents of the DE pair with the HL pair (XCHG) and to load and store the contents of the HL pair in two successive memory bytes (LHLD address and SHLD address).

Register pairs may either be pushed into and down the stack with PUSH rp, or popped (moved) from or off the stack into a register pair with POP rp. For stack operations, the register pair (rp) is specified as PSW (accumulator and status registers), D (DE), B (BC), or H (HL).

Registers, or register pairs are initialised with data by immediate addressing instructions. The MVI d data instruction causes the data in the 2-byte instruction to be loaded into the destination indicated by d (A, B, C, D, E, H, L or M). The LXI rp value causes 16 bits to be loaded into the register pair specified, which can include SP for stack pointer.

Arithmetic instructions

The arithmetic instructions that are used to add or subtract from the contents of the '8080A's accumulator (where the result is also held) are shown in *figure 5a*.

These instructions include: addition (ADD); addition with carry (ADC); addition of data from the instructions (ADI data and ACI data); subtractions (SUB); subtraction with borrow (SBB); and subtraction of data in the instruction (SUI data and SBI data).

5

a) Addition instructions

ADD	r	Add contents of register r to A register and place sum in A; r can be A,B,C,D,E,H,L or M
ADC	r	Add contents of register r plus the carry to the A register and place the results in A
ADI	data	Add data in instruction to A and place sum in
ACI	data	Add data in instruction plus carry to A and place sum in A
DAD	rp	Add 16 bits in register pair indicated to contents of HL pair and place sum in HL. rp can be B,D or H
INR	r	Add one to the contents of register r
INX	rp	Add one to the 16 bit contents of the register pair
DAA		Adjust the result of the previous addition so that both 4-bit codes in the A register are correct BCD codes

Subtraction instructions

SUB	r	Subtract contents of register r from A and place result in A
SBB	r	Subtract contents of register r and carry from contents of A and place result in A
SUI	data	Subtract data in instruction from contents of A and place result in A
SBI	data	Subtract data in instruction and carry from contents of A and place result in A
DCR	r	Subtract 1 from contents of the register r
DCX	rp	Subtract 1 from the 16-bit contents of the register pair

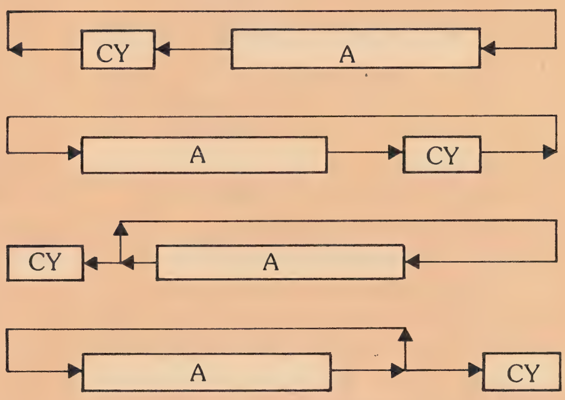
b) Accumulator rotation instructions

RAL

RAR

RLC

RRC



5. TMS8080A arithmetic instructions.

The increment register (INR), decrement register (DCR), increment register pair (INX) and decrement register pair (DCX) instructions are also a form of addition and subtraction, since one is added to or subtracted from the operand data.

The DAD rp instruction provides the addition of 16 bits from the BC, DE or HL pairs to the 16 bits contained in the HL pair with the result stored in the HL pair. The DAA (decimal adjust) instruction provides BCD addition.

The four rotate instructions shown in *figure 5b* – RAL, RAR, RLC, RRC – are also a form of arithmetic operation. The RAR (rotate right) and RAL (rotate left) rotate through the carry to allow for multi-byte shifts.

Logical operations

The basic AND, OR, Exclusive OR and complement operations are performed on

6. Logical instructions for the 8-bit TMS8080A.

6

AND instructions

ANA	r	The contents of the register r are ANDed with the contents of the A register; results to the A register. R may be A,B,C,D,E,H,L or M
ANI	data	The data in the instruction is ANDed with the contents of the A register with the results to the A register

OR instructions

ORA	r	The contents of the r register are ORed with the contents of the A register and the results to the A register
ORI	data	The data in the instruction is ORed with the contents of the A register with the results to the A register

Exlusive OR instructions

XRA	r	The contents of the register r are exclusive ORed with the contents of the A register; results to the A register
XRI	data	The data in the instruction is exclusive ORed with the contents of the A register; results to the A register

Complement

CMA		Complement the contents of A register, results to A register

7. Comparison, branch and miscellaneous instructions for the TMS8080A.

7

Comparison instructions

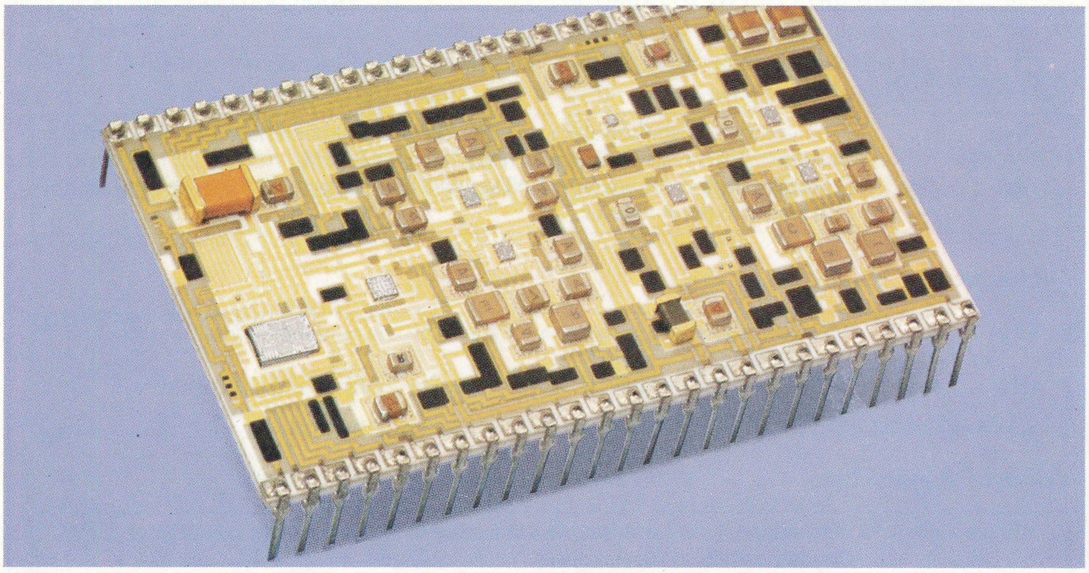
CMP	r	The contents of the register r are subtracted from the contents of the accumulator affecting the status bits. r can be A,B,C,D,E,H,L or M
CPI	data	The data in the instruction is subtracted from the contents of the A register, affecting only the status bits

Branch instructions

JMP	address	The address in the instruction is loaded into the program counter
Jcond	address	If the condition is true, the address in the instruction is loaded into the program counter. cond is Z (zero), NZ (not zero), C (carry), NC (no carry), P (plus), M (minus), PO (odd parity) or PE (even parity)
CALL	address	The address is loaded into the program counter and the old program counter value for the next instruction is saved on the stack
Ccond	address	A subroutine CALL operation occurs if the condition is met. Same condition possibilities as the Jcond
RET		The top of the stack is sent to the program counter
Rcond		If the condition is met, the top of the stack is sent to the program counter
RST	n	The program counter value for the next instruction is saved in the stack and the program counter is loaded with n x 8
PCHL		The program counter is loaded with the contents of the HL register pair

Miscellaneous instructions

EI	Enable the interrupt signal
DI	Disable the interrupt signal
HLT	Halt the processor; wait for an interrupt or RESET
NOP	No operation; time delay of 4 x clock period
CMC	Complement the carry flip-flop
STC	Set the carry flip-flop



Left: a hybrid circuit used in telecommunications applications. (Photo: Dupont).

the accumulator's contents by the instructions shown in *figure 6*. The results are also stored in the accumulator.

The two operand instructions allow the non-accumulator operand to be A, B, C, D, E, H, L, M or instruction data. The mnemonics are ANA (AND), ANI (AND immediate), XRA (Exclusive OR), XRI (exclusive OR immediate), ORA (OR), ORI (OR immediate) and CMA (complement the accumulator). With the immediate operations, the data used is contained in the second byte of the instruction code.

Comparison instructions

All the '8080A's comparisons are arithmetic. The data in the accumulator is compared to data in the instruction (CPI) or in one of the processor registers, or in the memory location addressed by the contents of the HL register pair (the CMP instruction).

In both cases, the data specified by the operand is subtracted from the accumulator and the status bits are affected. Neither item of data is affected by the comparison operation. It is left to the conditional branch instructions to test the results of the comparisons.

Branch instructions

The branch instructions indicated in *figure* 7, provide unconditional branching with the JNZ (not zero), JZ (zero), JNC (no carry), JC (carry), JP (positive), JM (minus), JPO (odd parity) and JPE (even parity).

The subroutine calls can similarly be unconditional (CALL) or conditional (CNZ, CZ, CNC, CC, CP, CM, CPO or CPE). The subroutine returns can be unconditional (RET) or conditional (RZ, RNZ, RNC, RC, RP, RM, RPO or RPE).

The restart instruction (RST n), previously discussed under timing and control, is also a subroutine jump to address n∗8 (meaning n x 8), where n is a value from 0 to 7 as specified by the restart instruction (RST n). The PCHL is an indirect address unconditional branch since the contents of the HL register pair are loaded into the program counter to cause the jump.

Miscellaneous instructions

The '8080A also has some miscellaneous instructions (*figure* 7).

The enable interrupt system (EI) and disable interrupt system (DI) instructions have already been discussed. The CMC is used to complement the carry flip-flop and the STC is used to set the carry flip-flop. The HLT (halt) instruction causes the microprocessor to stop executing instructions and await a reset or an interrupt before continuing operations. The NOP, or no operation instruction, is inserted to use up time while waiting for some timer to end or some other event to occur.

This instruction set is relatively complete, and allows the TMS8080A to be used in a wide variety of applications with relative ease.

(continued in part 47)

TEST YOUR PROGRESS
with the
I.T.E.C. QUIZ

COMMUNICATIONS – 16

1. Recommendations X.25:

a Specifies the interface between packet terminals and data circuit terminating equipment on packet switched networks
b Specifies the DTE/DCE interface on X.25 networks
c a and b
d None of these

2. The X-series of CCITT recommendations concerns:

a Data communications over digital networks
b Data communications over only packet switched networks
c Analogue communications over X.25 networks
d What happens when a number is dialled from a telephone terminal
e a and b
f All of these
g None of these

3. All DTE connected to an X.25 network must be packet terminals.

True or False?

4. The lowest level of communications protocol specifies:

a How logical sessions are maintained even though electrical connections are not
b The electrical and mechanical interfaces
c Communications via a gateway between two data networks
d a and b
e All of these
f None of these

5. Any level of protocol is completely independent of all other levels.

True or False?

6. The transport service protocols:

a Consist of the physical, data-link, network and transport layer protocols
b Consist of all the first four level protocols
c Specify how data is transmitted from one DTE to another
d Pay no regard to the meaning of the data manipulated
e b and c
f All of these
g None of these

7. Management of network resources is the province of the:

a Transport service protocols
b Session layer
c Sixth level
d b and c
e All of these
f None of these

8. High level protocols:

a Are used to define how data is manipulated from DTE to DTE
b Are formed from protocols of the transport layer, the presentation layer, and the application layer
c Are not concerned with electrical connections between each DTE and DCE
d b and c
e All of these
f None of these

BASIC REFRESHERS

1. Filters may be constructed from:

a Passive linear networks
b Active, operational amplifier networks
c Digital circuits
d b and c
e All of these
f None of these

2. A non-recursive filter is:

a A digital filter
b A finite impulse response filter
c Made from electronic components
d a and b
e All of these
f None of these

3. A transmission line is:

a Any conductor
b Any conductor used to transmit electrical or electromagnetic energy
c A form of telescopic aerial
d a and b
e b and c
f All of these
g None of these

4. Any transmission line may be modelled by:

a A piece of string
b A metal conductor
c An impedance
d A capacitor and an inductor
e All of these
f None of these

5. The characteristic resistance of a transmission line is given by:

a $\sqrt{\frac{L}{C}}$
b $\frac{2\pi fL}{R_o}$
c $2\pi fCR_o$
d $\sqrt{\frac{C}{R}}$
e All of these
f None of these

6. In a correctly terminated transmission line:

a The waveform supplied to the line appears to travel along it
b The waveform supplied to the line must be a sine wave
c A short circuit appears at the termination
d An open circuit appears at the termination
e a and b
f All of these

Answers to last week's quiz

COMMUNICATIONS – 15
1 f
2 c
3 a
4 True
5 d

MICROPROCESSORS – 9
1 c
2 a
3 b

MICROPROCESSORS – 10
1 d
2 b
3 f

COMING IN PART 47

Microprocessors 11 continues with a detailed look at both the hardware design and program development for a specific **application for the 8-bit TMS8080A**.

The second of two articles on protocols – *Communications 17* – examines the **Packet SwitchStream protocol, X.25**.

Computers & Society 15 discusses **information technology**.

What are the advantages of using a **16-bit microprocessor?** Find out in *Microprocessors 12*.

PLUS: *Basic Theory Refresher* – looking at **pulses on transmission lines**.